

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7679301号
(P7679301)

(45)発行日 令和7年5月19日(2025.5.19)

(24)登録日 令和7年5月9日(2025.5.9)

(51)国際特許分類		F I	
H 0 4 N	19/50 (2014.01)	H 0 4 N	19/50
H 0 4 N	19/102 (2014.01)	H 0 4 N	19/102
H 0 4 N	19/136 (2014.01)	H 0 4 N	19/136
H 0 4 N	19/186 (2014.01)	H 0 4 N	19/186
H 0 4 N	19/70 (2014.01)	H 0 4 N	19/70
請求項の数 13 (全43頁)			
(21)出願番号	特願2021-547465(P2021-547465)	(73)特許権者	511050697
(86)(22)出願日	令和2年2月28日(2020.2.28)		アリババ グループ ホウルディング リ
(65)公表番号	特表2022-523925(P2022-523925		ミテッド
	A)		英国領ケイマン諸島 グランド ケイマン
(43)公表日	令和4年4月27日(2022.4.27)		ジョージ タウン ピーオーボックス 8
(86)国際出願番号	PCT/US2020/020513		4 7 ワン キャピタル プレイス フォー
(87)国際公開番号	WO2020/180737		ス フロア
(87)国際公開日	令和2年9月10日(2020.9.10)	(74)代理人	100079108
審査請求日	令和5年2月27日(2023.2.27)		弁理士 稲葉 良幸
(31)優先権主張番号	62/813,728	(74)代理人	100109346
(32)優先日	平成31年3月4日(2019.3.4)		弁理士 大貫 敏史
(33)優先権主張国・地域又は機関		(74)代理人	100117189
	米国(US)		弁理士 江口 昭彦
(31)優先権主張番号	62/817,546	(74)代理人	100134120
(32)優先日	平成31年3月12日(2019.3.12)		弁理士 内藤 和彦
最終頁に続く		最終頁に続く	

(54)【発明の名称】 映像コンテンツを処理するための方法及びシステム

(57)【特許請求の範囲】

【請求項1】

ビットストリームを復号して映像ストリームの1つまたは複数のピクチャを出力する方法であって、

ビットストリームを受信すること、及び

前記ビットストリームのコード化された情報を使用して、1つ又は複数のピクチャを復号すること

を含み、

前記復号することが、

ピクチャに関連する複数のルマサンプルを再構築すること、及び

前記ピクチャに関連するクロマブロックを再構築すること

を含み、

前記クロマブロックを再構築することが、

前記クロマブロックが非ゼロ残差を有するかどうかを判定すること、

前記クロマブロックが1つ又は複数の非ゼロクロマ残差を有するとの判定に応じて、再構築された前記複数のルマサンプルの平均値を決定し、前記クロマブロックを再構築する前に、前記平均値に基づいて前記クロマブロックの残差をスケールリングすること、及び

前記クロマブロックが非ゼロクロマ残差を有さないとの判定に応じて、再構築された前記複数のルマサンプルの前記平均値の前記決定をバイパスすること

を含む、

方法。

【請求項 2】

再構築された前記複数のルマサンプルの前記平均値に基づいて前記クロマブロックの前記残差をスケールリングすることが、

再構築された前記複数のルマサンプルの前記平均値に基づいてクロマスケール因子を決定すること、及び

前記クロマブロックの前記残差に前記クロマスケール因子を適用すること

を含む、請求項 1 に記載の方法。

【請求項 3】

前記クロマブロックが非ゼロ残差を有するかどうか、前記クロマブロックに関連するコード化ブロックフラグの値に基づいて判定される、請求項 1 に記載の方法。

10

【請求項 4】

前記復号することが、

前記コード化ブロックフラグの前記値が 1 に等しいことに応じて、前記クロマブロックが 1 つ又は複数の非ゼロ残差を有すると判定すること

を更に含む、請求項 3 に記載の方法。

【請求項 5】

前記復号することが、

前記コード化ブロックフラグの前記値が 0 に等しいことに応じて、前記クロマブロックが非ゼロ残差を有しないと判定すること

を更に含む、請求項 3 に記載の方法。

20

【請求項 6】

前記クロマブロックが非ゼロ残差を有するかどうか、

前記クロマブロックの第 1 のクロマ成分に関連する第 1 のコード化ブロックフラグの値、及び

前記クロマブロックの第 2 のクロマ成分に関連する第 2 のコード化ブロックフラグの値に基づいて判定される、請求項 1 に記載の方法。

【請求項 7】

前記復号することが、

前記第 1 のコード化ブロックフラグの前記値、又は前記第 2 のコード化ブロックフラグの前記値の少なくとも一方が 1 に等しいことに応じて、前記クロマブロックが 1 つ又は複数の非ゼロ残差を有すると判定すること

を更に含む、請求項 6 に記載の方法。

30

【請求項 8】

前記復号することが、

前記第 1 のコード化ブロックフラグの前記値、及び前記第 2 のコード化ブロックフラグの前記値の両方が 0 に等しいことに応じて、前記クロマブロックが非ゼロ残差を有しないと判定すること

を更に含む、請求項 6 に記載の方法。

【請求項 9】

1 組の命令を記憶する 1 つ又は複数のメモリと、

1 つ又は複数のプロセッサと、

を含む機器であって、

前記 1 つ又は複数のプロセッサが、

ピクチャに関連する複数のルマサンプルを再構築すること、及び

前記ピクチャに関連するクロマブロックを再構築すること

を前記機器に行わせるために前記 1 組の命令を実行するように構成され、

前記クロマブロックを再構築する際、

前記クロマブロックが非ゼロ残差を有するかどうかを判定すること、

前記クロマブロックが 1 つ又は複数の非ゼロクロマ残差を有するとの判定に応じて、

40

50

再構築された前記複数のルマサンプルの平均値を決定し、前記クロマブロックを再構築する前に、前記平均値に基づいて前記クロマブロックの残差をスケールリングすること、及び

前記クロマブロックが非ゼロクロマ残差を有さないとの判定に応じて、再構築された前記複数のルマサンプルの前記平均値の前記決定をバイパスすること

を前記機器に更に行わせるために前記 1 組の命令を実行するように前記 1 つ又は複数のプロセッサが構成される、

機器。

【請求項 10】

前記 1 つ又は複数のプロセッサが、

再構築された前記複数のルマサンプルの前記平均値に基づいてクロマスケール因子を決定すること、及び

前記クロマブロックの前記残差に前記クロマスケール因子を適用すること

を前記機器に更に行わせるために前記 1 組の命令を実行するように構成される、請求項 9 に記載の機器。

【請求項 11】

前記 1 つ又は複数のプロセッサが、

前記クロマブロックに関連するコード化ブロックフラグの値に基づいて、前記クロマブロックが非ゼロ残差を有するかどうかを判定すること

を前記機器に更に行わせるために前記 1 組の命令を実行するように構成される、請求項 9 に記載の機器。

【請求項 12】

前記 1 つ又は複数のプロセッサが、

前記コード化ブロックフラグの前記値が 1 に等しいことに応じて、前記クロマブロックが 1 つ又は複数の非ゼロ残差を有すると判定すること

を前記機器に更に行わせるために前記 1 組の命令を実行するように構成される、請求項 11 に記載の機器。

【請求項 13】

前記 1 つ又は複数のプロセッサが、

前記コード化ブロックフラグの前記値が 0 に等しいことに応じて、前記クロマブロックが非ゼロ残差を有しないと判定すること

を前記機器に更に行わせるために前記 1 組の命令を実行するように構成される、請求項 11 に記載の機器。

【発明の詳細な説明】

【技術分野】

【0001】

関連出願の相互参照

[0001] 本開示は、どちらも参照によりその全体を本明細書に援用する、2019年3月4日に出願された米国仮特許出願第62/813,728号、及び2019年3月12日に出願された米国仮特許出願第62/817,546号の優先権の利益を主張する。

【0002】

技術分野

[0002] 本開示は一般に映像処理に関し、より詳細にはクロマスケールリングによるインテリジェントマッピングを行うための方法及びシステムに関する。

【背景技術】

【0003】

背景

[0003] 映像コード化システムは、例えば消費される記憶空間を減らすべく又はかかる信号に関連する伝送帯域幅の消費量を減らすべく、デジタル映像信号を圧縮するためにしばしば使用される。オンライン映像ストリーミング、テレビ会議、又は映像監視等の映像圧縮の様々な応用において（例えば1920x1080ピクセルの解像度を有する）高精

10

20

30

40

50

細度（HD）映像の人気の高まるにつれ、映像データの圧縮効率を高めることができる映像コード化ツールを開発することが絶えず求められる。

【0004】

【0004】例えば映像監視の応用は多くの応用シナリオ（例えばセキュリティ、交通、環境のモニタリング等）において一層且つ広範に使用されており、監視装置の台数及び解像度が急激に増加している。多くの映像監視の応用シナリオはより多くの情報を捕捉するためにHD映像をユーザに提供することを選び、HD映像はかかる情報を捕捉するために1フレーム当たりより多くのピクセルを有する。しかしHD映像ビットストリームは、伝送のための高帯域幅及び記憶のための大きい空間を要求する高ビットレートを有し得る。例えば平均的な1920×1080の解像度を有する監視映像ストリームは、リアルタイム伝送のために4Mbpsもの帯域幅を必要とし得る。更に、映像監視は一般に7日間×24時間、常時監視を行い、そのことは映像データを記憶する場合は記憶システムの能力を著しく試し得る。従って、HD映像の高帯域幅及び大きい記憶域に対する需要は映像監視におけるHD映像の大規模な展開に対する主な制限になっている。

10

【発明の概要】

【課題を解決するための手段】

【0005】

開示の概要

【0005】本開示の実施形態は映像コンテンツを処理するための方法を提供する。この方法は、ピクチャに関連するクロマブロック及びルマブロックを受信すること、ルマブロックに関連するルマスケール情報を決定すること、ルマスケール情報に基づいてクロマスケール因子を決定すること、及びクロマスケール因子を使用してクロマブロックを処理することを含み得る。

20

【0006】

【0006】本開示の実施形態は映像コンテンツを処理するための機器を提供する。この機器は、1組の命令を記憶するメモリと、メモリに結合され、ピクチャに関連するクロマブロック及びルマブロックを受信すること、ルマブロックに関連するルマスケール情報を決定すること、ルマスケール情報に基づいてクロマスケール因子を決定すること、及びクロマスケール因子を使用してクロマブロックを処理することを機器に行わせるために、1組の命令を実行するように構成されるプロセッサとを含み得る。

30

【0007】

【0007】本開示の実施形態は、映像コンテンツを処理するための方法を装置に行わせるために、装置の1つ又は複数のプロセッサによって実行可能な1組の命令を記憶する非一時的コンピュータ可読記憶媒体を提供する。その方法は、ピクチャに関連するクロマブロック及びルマブロックを受信すること、ルマブロックに関連するルマスケール情報を決定すること、ルマスケール情報に基づいてクロマスケール因子を決定すること、及びクロマスケール因子を使用してクロマブロックを処理することを含む。

【0008】

図面の簡単な説明

【0008】本開示の実施形態及び様々な態様を以下の詳細な説明及び添付図面の中で示す。図中に示す様々な特徴は縮尺通り描かれていない。

40

【図面の簡単な説明】

【0009】

【図1】【0009】本開示の一部の実施形態による映像シーケンスの一例の構造を示す。

【図2A】【0010】本開示の一部の実施形態による符号化プロセスの一例の概略図を示す。

【図2B】【0011】本開示の一部の実施形態による符号化プロセスの別の例の概略図を示す。

。

【図3A】【0012】本開示の一部の実施形態による復号プロセスの一例の概略図を示す。

【図3B】【0013】本開示の一部の実施形態による復号プロセスの別の例の概略図を示す。

【図4】【0014】本開示の一部の実施形態による、映像を符号化又は復号するための機器の

50

一例のブロック図を示す。

【図 5】[0015]本開示の一部の実施形態による、例示的なクロマスケールングによるルママッピング (LMCS: luma mapping with chroma scaling) プロセスの概略図を示す。

【図 6】[0016]本開示の一部の実施形態による、LMCS 区分線形モデルのためのタイルグループレベルの構文表を示す。

【図 7】[0017]本開示の一部の実施形態による、LMCS 区分線形モデルのための別のタイルグループレベルの構文表を示す。

【図 8】[0018]本開示の一部の実施形態による、コード化ツリー単位の構文構造の表である。

【図 9】[0019]本開示の一部の実施形態による、デュアルツリー分割の構文構造の表である。

【図 10】[0020]本開示の一部の実施形態による、ルマ予測ブロックの平均化を単純化する一例を示す。

【図 11】[0021]本開示の一部の実施形態による、コード化ツリー単位の構文構造の表である。

【図 12】[0022]本開示の一部の実施形態による、タイルグループレベルにおける LMCS 区分線形モデルの修正されたシグナリングに関する構文要素の表である。

【図 13】[0023]本開示の一部の実施形態による、映像コンテンツを処理するための方法の流れ図である。

【図 14】[0024]本開示の一部の実施形態による、映像コンテンツを処理するための方法の流れ図である。

【図 15】[0025]本開示の一部の実施形態による、映像コンテンツを処理するための別の方法の流れ図である。

【図 16】[0026]本開示の一部の実施形態による、映像コンテンツを処理するための別の方法の流れ図である。

【発明を実施するための形態】

【0010】

詳細な説明

[0027] 次に、その例を添付図面に示す例示的实施形態を詳細に参照する。以下の説明は添付図面を参照し、添付図面では他に指示がない限り異なる図中の同じ数字が同じ又は同様の要素を表す。例示的实施形態についての以下の説明の中で記載する実装形態は本発明と合致する全ての実装形態を表すわけではない。むしろ、それらは添付の特許請求の範囲の中で列挙する本発明に関係する態様と合致する機器及び方法の例に過ぎない。別段の定めがない限り、「又は、若しくは (or)」という語は実行不可能な場合を除いてあり得る全ての組み合わせを包含する。例えば或る構成要素が A 又は B を含む得ると述べた場合、別段の定めがない限り又は実行不可能でない限り、その構成要素は A、若しくは B、又は A 及び B を含むことができる。第 2 の例として、或る構成要素が A、B、又は C を含む得ると述べた場合、別段の定めがない限り又は実行不可能でない限り、その構成要素は A、若しくは B、若しくは C、又は A 及び B、又は A 及び C、又は B 及び C、又は A 及び B 及び C を含むことができる。

【0011】

[0028] 映像とは、視覚的情報を記憶するために時系列順に配置される 1 組の静止ピクチャ (又は「フレーム」) である。それらのピクチャを時系列順に捕捉し記憶するために映像捕捉装置 (例えばカメラ) を使用することができ、かかるピクチャを時系列順に表示するために映像再生装置 (例えばテレビ、コンピュータ、スマートフォン、タブレットコンピュータ、ビデオプレーヤ、又は表示機能を有する任意のエンドユーザ端末) を使用することができる。更に一部の応用では、監視、会議、又は生放送等のために映像捕捉装置が捕捉映像を映像再生装置 (例えばモニタを有するコンピュータ) にリアルタイムで伝送することができる。

10

20

30

40

50

【 0 0 1 2 】

[0029] かかる応用が必要とする記憶空間及び伝送帯域幅を減らすために、映像を記憶及び伝送前に圧縮し、表示前に解凍することができる。この圧縮及び解凍はプロセッサ（例えば汎用コンピュータのプロセッサ）又は専用ハードウェアによって実行されるソフトウェアによって実装することができる。圧縮用のモジュールを一般に「符号器」と呼び、解凍用のモジュールを一般に「復号器」と呼ぶ。符号器及び復号器はまとめて「コーデック」と呼ぶことができる。符号器及び復号器は様々な適切なハードウェア、ソフトウェア、又はその組み合わせとして実装することができる。例えば符号器及び復号器のハードウェア実装は、1つ又は複数のマイクロプロセッサ、デジタル信号プロセッサ（DSP）、特定用途向け集積回路（ASIC）、書換可能ゲートアレイ（FPGA）、ディスクリートロジック、又はその任意の組み合わせ等の回路を含み得る。符号器及び復号器のソフトウェア実装は、プログラムコード、コンピュータ実行可能命令、ファームウェア、又はコンピュータ可読媒体内に固定される任意の適切なコンピュータによって実装されるアルゴリズム若しくはプロセスを含み得る。映像の圧縮及び解凍はMPEG-1、MPEG-2、MPEG-4、H.26xシリーズ等の様々なアルゴリズム又は規格によって実装され得る。一部の応用では、コーデックが第1のコード化規格から映像を解凍し、第2のコード化規格を使用して解凍済みの映像を再圧縮することができ、その場合コーデックを「トランスコーダ」と呼ぶことができる。

10

【 0 0 1 3 】

[0030] 映像符号化プロセスは、ピクチャを再構築するために使用可能な有用な情報を識別し保つことができ、再構築に重要でない情報を無視することができる。無視された重要でない情報を完全に再構築できない場合、かかる符号化プロセスは「非可逆」と呼ぶことができる。さもないければ、かかる符号化プロセスは「可逆」と呼ぶことができる。殆どの符号化プロセスは非可逆であり、これは必要な記憶空間及び伝送帯域幅を減らすためのトレードオフである。

20

【 0 0 1 4 】

[0031] 符号化されているピクチャ（「現ピクチャ」と呼ぶ）の有用な情報は、参照ピクチャ（例えば過去に符号化され再構築されたピクチャ）に対する変化を含む。かかる変化は、ピクセルの位置変化、光度変化、又は色変化を含むことができ、そのうちの位置変化が最も関係している。オブジェクトを表すピクセル群の位置変化は、参照ピクチャと現ピクチャとの間のオブジェクトの動きを反映し得る。

30

【 0 0 1 5 】

[0032] 別のピクチャを参照することなしにコード化されるピクチャ（即ちかかるピクチャが自らの参照ピクチャである）を「Iピクチャ」と呼ぶ。参照ピクチャとして過去のピクチャを使用してコード化されるピクチャを「Pピクチャ」と呼ぶ。参照ピクチャとして過去のピクチャ及び将来のピクチャの両方を使用してコード化される（即ち参照が「双方向」である）ピクチャを「Bピクチャ」と呼ぶ。

【 0 0 1 6 】

[0033] 先に述べたように、HD映像を使用する映像監視は高帯域幅及び大きい記憶域の需要の課題に直面する。この課題に対処するために符号化映像のビットレートを下げることができる。Iピクチャ、Pピクチャ、及びBピクチャのうちIピクチャが最も高いビットレートを有する。殆どの監視映像の背景はほぼ静的なので、符号化映像の全体的なビットレートを下げる1つのやり方は、映像の符号化のためにより少ないIピクチャを使用することであり得る。

40

【 0 0 1 7 】

[0034] しかし、符号化映像内でIピクチャは一般に主要ではないので、Iピクチャをより少なく使用する改善策は些細なことであり得る。例えば典型的な映像ビットストリームでは、Iピクチャ、Bピクチャ、及びPピクチャの比率が1:20:9の場合があり、Iピクチャが総ビットレートの10%未満を占めることがある。換言すれば、かかる例では全てのIピクチャを除去しても低減されるビットレートは10%に過ぎない可能性があ

50

る。

【 0 0 1 8 】

[0035] 本開示は、映像監視のための特性ベースの映像処理のための方法、機器、及びシステムを提供する。本明細書の「特性」は、ピクチャ内の映像コンテンツに関連するコンテンツ特性、ピクチャを符号化すること又は復号することの動き推定に関連する動き特性、又はその両方を指す。例えばコンテンツ特性は映像シーケンスの1つ又は複数の連続ピクチャ内のピクセルとすることができ、ピクセルはピクチャ内のオブジェクト、シーン、又は環境事象の少なくとも1つに関係する。別の例では動き特性が映像のコード化プロセスに関係する情報を含むことができ、その例は後で詳述する。

【 0 0 1 9 】

[0036] 本開示では、映像シーケンスのピクチャを符号化するとき、映像シーケンスのピクチャの1つ又は複数の特性を検出し分類するために特性分類器を使用することができ、特性の様々なクラスを様々な優先レベルに関連付けることができ、かかる優先レベルは符号化のための様々なビットレートに更に関連付けられる。様々な優先レベルは符号化のための様々なパラメータセットに関連付けることができ、それにより異なる符号化品質レベルがもたらされ得る。優先レベルが高ければ高いほど、その関連パラメータセットがもたらし得る映像の質が高くなる。このような特性ベースの映像処理により、著しい情報損失を引き起こすことなく監視映像についてビットレートを大幅に下げることができる。加えて、本開示の実施形態は様々な応用シナリオ（例えばセキュリティ、交通、環境のモニタリング等）について優先レベルとパラメータセットとの間の対応する関係をカスタマイズし、それにより映像のコード化品質を大幅に改善し、帯域幅及び記憶域のコストを大幅に低減することができる。

【 0 0 2 0 】

[0037] 図1は、本開示の一部の実施形態による映像シーケンス100の一例の構造を示す。映像シーケンス100は、生中継映像又は捕捉されアーカイブされている映像とすることができる。映像100は現実の映像、コンピュータによって生成される映像（例えばコンピュータゲーム映像）、又はその組み合わせ（例えば拡張現実効果を有する現実の映像）であり得る。映像シーケンス100は、映像捕捉装置（例えばカメラ）、過去に捕捉された映像を含む映像アーカイブ（例えば記憶装置内に記憶される映像ファイル）、又は映像コンテンツプロバイダから映像を受信するための映像フィードインタフェース（例えば映像ブロードキャストトランシーバ）から入力され得る。

【 0 0 2 1 】

[0038] 図1に示すように、映像シーケンス100は、ピクチャ102、104、106、及び108を含むタイムライン沿いに時間的に配置される一連のピクチャを含み得る。ピクチャ102～106は連続的であり、ピクチャ106とピクチャ108との間には更に多くのピクチャがある。図1ではピクチャ102がIピクチャであり、その参照ピクチャはピクチャ102自体である。ピクチャ104はPピクチャであり、矢印によって示すようにその参照ピクチャはピクチャ102である。ピクチャ106はBピクチャであり、矢印によって示すようにその参照ピクチャはピクチャ104及び108である。一部の実施形態では、ピクチャ（例えばピクチャ104）の参照ピクチャはそのピクチャのすぐ前に又はすぐ後になくてもよい。例えばピクチャ104の参照ピクチャは、ピクチャ102に先行するピクチャであり得る。ピクチャ102～106の参照ピクチャは例に過ぎず、本開示は参照ピクチャの実施形態を図1に示す例として限定しないことに留意すべきである。

【 0 0 2 2 】

[0039] 典型的には、映像コーデックは全ピクチャを一度に符号化又は復号することではなく、それはかかるタスクが計算的に複雑だからである。むしろ映像コーデックはピクチャを基本セグメントに分割し、ピクチャをセグメントごとに符号化又は復号することができる。本開示ではそのような基本セグメントを基本処理単位（「BPU」）と呼ぶ。例えば図1の構造110は、映像シーケンス100のピクチャ（例えばピクチャ102～10

8の何れか)の構造の一例を示す。構造110ではピクチャが4×4の基本処理単位に分けられており、その境界が破線で示されている。一部の実施形態では、基本処理単位は一部の映像コード化規格(例えばMPEGファミリー、H.261、H.263、又はH.264/AVC)内の「マクロブロック」と呼ぶことができ、他の一部の映像コード化規格(例えばH.265/HEVC又はH.266/VVC)内の「コード化ツリー単位」(「CTU」と呼ぶことができる。128×128、64×64、32×32、16×16、4×8、16×32、又はピクセルのあらゆる任意の形状及びサイズ等、基本処理単位はピクチャ内で可変サイズを有することができる。基本処理単位のサイズ及び形状は、コード化の効率及び基本処理単位内で保とうとする詳細度のバランスに基づいてピクチャについて選択することができる。

10

【0023】

[0040] 基本処理単位は、コンピュータメモリ内に(例えば映像フレームバッファ内に)記憶される様々な種類の映像データ群を含み得る論理単位であり得る。例えばカラーピクチャの基本処理単位は、無彩色の輝度情報を表すルマ成分(Y)、色情報を表す1つ又は複数のクロマ成分(例えばCb及びCr)、並びにルマ成分及びクロマ成分が同じサイズを有し得る基本処理単位の関連構文要素を含むことができる。一部の映像コード化規格(例えばH.265/HEVC又はH.266/VVC)では、ルマ成分及びクロマ成分が「コード化ツリーブロック」(「CTB」と呼ばれ得る。基本処理単位に対して行われる如何なる操作も、そのルマ成分及びクロマ成分のそれぞれに対して繰り返し行うことができる。

20

【0024】

[0041] 映像のコード化は複数の操作段階を有し、その例を図2A~図2B及び図3A~図3Bの中で詳述する。それぞれの段階について、基本処理単位のサイズは依然として処理するのに大き過ぎる場合があり、従って本開示で「基本処理副単位」と呼ぶセグメントに更に分けることができる。一部の実施形態では、基本処理副単位は一部の映像コード化規格(例えばMPEGファミリー、H.261、H.263、又はH.264/AVC)内の「ブロック」と呼ぶことができ、又は他の一部の映像コード化規格(例えばH.265/HEVC又はH.266/VVC)内の「コード化単位」(「CU」と呼ぶことができる。基本処理副単位は基本処理単位と同じ又はそれよりも小さいサイズを有し得る。基本処理単位と同様に基本処理副単位もコンピュータメモリ内に(例えば映像フレームバッファ内に)記憶される様々な種類の映像データ群(例えばY、Cb、Cr、及び関連構文要素)を含み得る論理単位である。基本処理副単位に対して行われる如何なる操作も、そのルマ成分及びクロマ成分のそれぞれに対して繰り返し行うことができる。処理の必要性に応じて、かかる分割は更なるレベルに対して行われ得ることに留意すべきである。様々な段階が様々な方式を使用して基本処理単位を分割できることに留意すべきである。

30

【0025】

[0042] 例えば(その一例を図2Bで詳述する)モード決定段階において、基本処理単位に対してどの予測モード(例えばイントラピクチャ予測又はインタピクチャ予測)を使用するのかを符号器が決定することができ、基本処理単位はかかる決定を下すには大き過ぎる場合がある。符号器は基本処理単位を複数の基本処理副単位(例えばH.265/HEVC又はH.266/VVCにあるCU)に分け、個々の基本処理副単位ごとに予測の種類を決定することができる。

40

【0026】

[0043] 別の例では、(その一例を図2Aに詳述する)予測段階において、符号器は基本処理副単位(例えばCU)のレベルにおいて予測操作を行うことができる。しかし一部の事例では、処理するのに基本処理副単位が依然として大き過ぎる場合がある。符号器は基本処理副単位をより小さいセグメント(例えばH.265/HEVC又はH.266/VVC内で「予測ブロック」又は「PB」と呼ばれる)に更に分けることができ、そのレベルにおいて予測操作を行うことができる。

【0027】

50

【0044】 別の例では、（その一例を図2Aに詳述する）変換段階において、符号器が残差基本処理副単位（例えばCU）に対する変換操作を行うことができる。しかし一部の事例では、処理するのに基本処理副単位が依然として大き過ぎる場合がある。符号器は基本処理副単位をより小さいセグメント（例えばH.265/HVC又はH.266/VVC内で「変換ブロック」又は「TB」と呼ばれる）に更に分けることができ、そのレベルにおいて変換操作を行うことができる。同じ基本処理副単位の分割方式は予測段階と変換段階とで異なり得ることに留意すべきである。例えばH.265/HVC又はH.266/VVCでは、同じCUの予測ブロック及び変換ブロックが異なるサイズ及び数を有し得る。

【0028】

10

【0045】 図1の構造110では、基本処理単位112が3×3の基本処理副単位に更に分けられており、その境界が点線で示されている。同じピクチャの異なる基本処理単位を異なる方式で基本処理副単位に分けることができる。

【0029】

【0046】 一部の実装形態では、映像の符号化及び復号に並列処理及び誤り耐性の機能を与えるために、ピクチャを処理用の領域に分けることができ、それによりピクチャの領域について、符号化又は復号プロセスがピクチャの他の任意の領域の情報に依存しないようにできる。換言すれば、ピクチャの各領域を独立に処理することができる。そうすることでコーデックはピクチャの異なる領域を並列に処理し、従ってコード化の効率を高めることができる。更に、領域のデータが処理内で破損し又はネットワーク伝送内で失われる場合、コーデックは破損し又は失われたデータに依存することなしに、同じピクチャの他の領域を正しく符号化又は復号することができ、従って誤り耐性の機能を提供する。一部の映像コード化規格では、ピクチャを異なる種類の領域に分割することができる。例えばH.265/HVC及びH.266/VVCは「スライス」及び「タイル」という2種類の領域を提供する。映像シーケンス100の様々なピクチャが、ピクチャを領域に分けるための様々な分割方式を有し得ることに留意すべきである。

20

【0030】

【0047】 例えば図1では、構造110が3つの領域114、116、及び118に分けられており、その境界が構造110内の実線として示されている。領域114は4個の基本処理単位を含む。領域116及び118のそれぞれは6個の基本処理単位を含む。図1の構造110の基本処理単位、基本処理副単位、及び領域は例に過ぎず、本開示はその実施形態を限定しないことに留意すべきである。

30

【0031】

【0048】 図2Aは、本開示の一部の実施形態による符号化プロセス200Aの一例の概略図を示す。符号器は、プロセス200Aに従って映像シーケンス202を映像ビットストリーム228へと符号化することができる。図1の映像シーケンス100と同様に、映像シーケンス202は時系列順に配置される1組のピクチャ（「元のピクチャ」と呼ぶ）を含み得る。図1の構造110と同様に、映像シーケンス202のそれぞれの元のピクチャは符号器によって基本処理単位、基本処理副単位、又は処理用の領域に分けられ得る。一部の実施形態では、符号器は映像シーケンス202のそれぞれの元のピクチャに関する基本処理単位のレベルにおいてプロセス200Aを実行することができる。例えば符号器はプロセス200Aを反復的なやり方で実行することができ、符号器はプロセス200Aの1回の反復の中で基本処理単位を符号化することができる。一部の実施形態では、符号器は映像シーケンス202のそれぞれの元のピクチャの領域（例えば領域114～118）についてプロセス200Aを並列に実行することができる。

40

【0032】

【0049】 図2Aでは、符号器は、映像シーケンス202の元のピクチャの基本処理単位（「元のBPU」と呼ぶ）を予測段階204にフィードして、予測データ206及び予測済みBPU208を生成することができる。符号器は、元のBPUから予測済みBPU208を減算して、残差BPU210を生成することができる。符号器は、残差BPU21

50

0 を変換段階 2 1 2 及び量子化段階 2 1 4 にフィードして、量子化済み変換係数 2 1 6 を生成することができる。符号器は、予測データ 2 0 6 及び量子化済み変換係数 2 1 6 をバイナリコード化段階 2 2 6 にフィードして、映像ビットストリーム 2 2 8 を生成することができる。構成要素 2 0 2、2 0 4、2 0 6、2 0 8、2 1 0、2 1 2、2 1 4、2 1 6、2 2 6、及び 2 2 8 は「順方向経路」と呼ぶことができる。プロセス 2 0 0 A の間、符号器は、量子化段階 2 1 4 の後で量子化済み変換係数 2 1 6 を逆量子化段階 2 1 8 及び逆変換段階 2 2 0 にフィードして、再構築済み残差 B P U 2 2 2 を生成することができる。符号器は、再構築済み残差 B P U 2 2 2 を予測済み B P U 2 0 8 に加えて、プロセス 2 0 0 A の次の反復の予測段階 2 0 4 に使用される予測基準 2 2 4 を生成することができる。プロセス 2 0 0 A の構成要素 2 1 8、2 2 0、2 2 2、及び 2 2 4 は「再構築経路」と呼ぶことができる。再構築経路は、符号器及び復号器の両方が予測に同じ参照データを使用することを確実にするために使用され得る。

10

【0033】

[0050] 符号器はプロセス 2 0 0 A を反復的に実行して、（順方向経路内で）元のピクチャのそれぞれの元の B P U を符号化し、（再構築経路内で）元のピクチャの次の元の B P U を符号化するための予測済み基準 2 2 4 を生成することができる。元のピクチャの全ての元の B P U を符号化した後、符号器は映像シーケンス 2 0 2 内の次のピクチャの符号化に進むことができる。

【0034】

[0051] プロセス 2 0 0 A を参照し、符号器は、映像捕捉装置（例えばカメラ）によって生成される映像シーケンス 2 0 2 を受信することができる。本明細書で使用する「受信（する）」という用語は、データを入力するために受信すること、入力すること、取得すること、取り出すこと、得ること、読み出すこと、アクセスすること、又は任意のやり方の任意のアクションを指すことができる。

20

【0035】

[0052] 現在の反復の予測段階 2 0 4 で、符号器は、元の B P U 及び予測基準 2 2 4 を受信し、予測操作を行って予測データ 2 0 6 及び予測済み B P U 2 0 8 を生成することができる。予測基準 2 2 4 は、プロセス 2 0 0 A の前の反復の再構築経路から生成され得る。予測段階 2 0 4 の目的は、予測データ 2 0 6 及び予測基準 2 2 4 から予測済み B P U 2 0 8 として元の B P U を再構築するために使用され得る予測データ 2 0 6 を抽出することによって、情報の冗長性を減らすことである。

30

【0036】

[0053] 理想的には、予測済み B P U 2 0 8 は元の B P U と同一であり得る。しかし、理想的でない予測及び再構築操作により、予測済み B P U 2 0 8 は概して元の B P U と僅かに異なる。そのような差を記録するために、符号器は、予測済み B P U 2 0 8 を生成した後でそれを元の B P U から減算して、残差 B P U 2 1 0 を生成することができる。例えば符号器は、予測済み B P U 2 0 8 のピクセルの値（例えばグレースケール値又は R G B 値）を元の B P U の対応するピクセルの値から減算することができる。元の B P U の対応するピクセルと予測済み B P U 2 0 8 との間のかかる減算の結果、残差 B P U 2 1 0 の各ピクセルは残差値を有し得る。元の B P U と比較し、予測データ 2 0 6 及び残差 B P U 2 1 0 はより少ないビットを有し得るが、品質を著しく損なうことなしに元の B P U を再構築するためにそれらを使用することができる。

40

【0037】

[0054] 残差 B P U 2 1 0 を更に圧縮するために、変換段階 2 1 2 で、符号器は残差 B P U 2 1 0 を 1 組の 2 次元「基底パターン」へと分解することによって、残差 B P U 2 1 0 の空間的冗長性を低減することができ、各基底パターンは「変換係数」に関連する。基底パターンは同じサイズ（例えば残差 B P U 2 1 0 のサイズ）を有することができる。それぞれの基底パターンは残差 B P U 2 1 0 の変動周波数（例えば輝度変動周波数）成分を表すことができる。基底パターンのどれも、他の任意の基底パターンの任意の組み合わせ（例えば線形結合）から再現することはできない。換言すれば、かかる分解により、残差

50

B P U 2 1 0 の変動を周波数領域内に分解することができる。かかる分解は関数の離散フーリエ変換に類似しており、基底パターンは離散フーリエ変換の基底関数（例えば三角関数）に類似しており、変換係数は基底関数に関連する係数に類似している。

【 0 0 3 8 】

[0055] 様々な変換アルゴリズムが様々な基底パターンを使用することができる。例えば離散コサイン変換、離散サイン変換等、変換段階 2 1 2 では様々な変換アルゴリズムを使用することができる。変換段階 2 1 2 における変換は可逆的である。つまり符号器は変換の逆操作（「逆変換」と呼ぶ）によって、残差 B P U 2 1 0 を復元することができる。例えば残差 B P U 2 1 0 のピクセルを復元するために、逆変換は、基底パターンの対応するピクセルの値に関連するそれぞれの係数で乗算し、積を加算して加重和をもたらすことであり得る。映像コード化規格では、符号器及び復号器の両方が同じ変換アルゴリズム（従って同じ基底パターン）を使用することができる。従って符号器は、復号器が符号器から基底パターンを受信することなしに、そこから残差 B P U 2 1 0 を再構築することができる変換係数だけを記録することができる。残差 B P U 2 1 0 と比較して変換係数の方が少ないビットを有し得るが、それらの変換係数は、品質を著しく損なうことなしに残差 B P U 2 1 0 を再構築するために使用され得る。従って残差 B P U 2 1 0 が更に圧縮される。

10

【 0 0 3 9 】

[0056] 符号器は、量子化段階 2 1 4 において変換係数を更に圧縮することができる。変換プロセスでは、様々な基底パターンが様々な変動周波数（例えば輝度変動周波数）を表すことができる。人間の目は概して低周波変動を認識するのが得意なので、符号器は復号の際の著しい品質劣化を引き起こすことなしに、高周波変動の情報を無視することができる。例えば量子化段階 2 1 4 において、符号器は、各変換係数を整数値（「量子化パラメータ」と呼ぶ）で除算し、商をその最近隣数に丸めることによって、量子化済み変換係数 2 1 6 を生成することができる。かかる操作の後、高周波基底パターンの一部の変換係数をゼロに変換することができ、低周波基底パターンの変換係数をより小さい整数に変換することができる。符号器はゼロ値の量子化済み変換係数 2 1 6 を無視することができ、それにより変換係数が更に圧縮される。量子化プロセスも可逆的であり、量子化済み変換係数 2 1 6 は、量子化の逆操作（「逆量子化」と呼ぶ）で変換係数に再構築することができる。

20

【 0 0 4 0 】

[0057] 符号器は、丸め操作でかかる除算の剰余を無視するので、量子化段階 2 1 4 は非可逆であり得る。典型的には、量子化段階 2 1 4 はプロセス 2 0 0 A 内で最大の情報損失に寄与し得る。情報損失が大きければ大きいほど、量子化済み変換係数 2 1 6 が必要とし得るビットが少なくなる。情報損失の様々なレベルを得るために、符号器は、量子化パラメータの様々な値又は量子化プロセスの他の任意のパラメータを使用することができる。

30

【 0 0 4 1 】

[0058] バイナリコード化段階 2 2 6 において、符号器は例えばエントロピコード化、可変長コード化、算術コード化、ハフマンコード化、コンテキスト適応バイナリ算術コード化、又は他の任意の可逆若しくは非可逆圧縮アルゴリズム等のバイナリコード化技法を使用し、予測データ 2 0 6 及び量子化済み変換係数 2 1 6 を符号化することができる。一部の実施形態では、予測データ 2 0 6 及び量子化済み変換係数 2 1 6 に加えて、符号器は、例えば予測段階 2 0 4 で使用される予測モード、予測操作のパラメータ、変換段階 2 1 2 の変換の種類、量子化プロセスのパラメータ（例えば量子化パラメータ）、符号器制御パラメータ（例えばビットレート制御パラメータ）等の他の情報をバイナリコード化段階 2 2 6 において符号化することができる。符号器は、バイナリコード化段階 2 2 6 の出力データを使用して映像ビットストリーム 2 2 8 を生成することができる。一部の実施形態では、映像ビットストリーム 2 2 8 をネットワーク伝送のために更にパケット化することができる。

40

【 0 0 4 2 】

[0059] プロセス 2 0 0 A の再構築経路を参照し、逆量子化段階 2 1 8 で、符号器は、

50

量子化済み変換係数 2 1 6 に対して逆量子化を行って再構築済み変換係数を生成することができる。逆変換段階 2 2 0 で、符号器は、再構築済み変換係数に基づいて再構築済み残差 B P U 2 2 2 を生成することができる。符号器は、再構築済み残差 B P U 2 2 2 を予測済み B P U 2 0 8 に加えて、プロセス 2 0 0 A の次の反復内で使用される予測基準 2 2 4 を生成することができる。

【 0 0 4 3 】

[0060] 映像シーケンス 2 0 2 を符号化するためにプロセス 2 0 0 A の他のパリエーションを使用できることに留意すべきである。一部の実施形態では、符号器がプロセス 2 0 0 A の段階を異なる順序で実行することができる。一部の実施形態では、プロセス 2 0 0 A の 1 つ又は複数の段階を単一の段階へと組み合わせることができる。一部の実施形態では、プロセス 2 0 0 A の単一の段階を複数の段階に分けることができる。例えば変換段階 2 1 2 と量子化段階 2 1 4 とを単一の段階へと組み合わせることができる。一部の実施形態では、プロセス 2 0 0 A が追加の段階を含み得る。一部の実施形態では、プロセス 2 0 0 A が図 2 A 内の 1 つ又は複数の段階を省くことができる。

【 0 0 4 4 】

[0061] 図 2 B は、本開示の一部の実施形態による符号化プロセスの別の例 2 0 0 B の概略図を示す。プロセス 2 0 0 B は、プロセス 2 0 0 A から修正され得る。例えばプロセス 2 0 0 B は、ハイブリッド映像コード化規格（例えば H . 2 6 x シリーズ）に準拠する符号器によって使用され得る。プロセス 2 0 0 A と比較し、プロセス 2 0 0 B の順方向経路はモード決定段階 2 3 0 を更に含み、予測段階 2 0 4 を空間的予測段階 2 0 4 2 及び時間的予測段階 2 0 4 4 に分ける。プロセス 2 0 0 B の再構築経路は、ループフィルタ段階 2 3 2 及びバッファ 2 3 4 を追加で含む。

【 0 0 4 5 】

[0062] 概して、予測技法は空間的予測及び時間的予測の 2 つの種類に分類することができる。空間的予測（例えばイントラピクチャ予測又は「イントラ予測」）は、現 B P U を予測するために、同じピクチャ内の既にコード化された 1 つ又は複数の隣接 B P U のピクセルを使用することができる。つまり空間的予測における予測基準 2 2 4 は、隣接 B P U を含み得る。空間的予測はピクチャの固有の空間的冗長性を減らすことができる。時間的予測（例えばインタピクチャ予測又は「インタ予測」）は、現 B P U を予測するために、既にコード化された 1 つ又は複数のピクチャの領域を使用することができる。つまり時間的予測における予測基準 2 2 4 は、コード化されたピクチャを含み得る。時間的予測はピクチャの固有の時間的冗長性を減らすことができる。

【 0 0 4 6 】

[0063] プロセス 2 0 0 B を参照し、順方向経路において、符号器は、空間的予測段階 2 0 4 2 及び時間的予測段階 2 0 4 4 で予測操作を行う。例えば空間的予測段階 2 0 4 2 で符号器はイントラ予測を行うことができる。符号化されているピクチャの元の B P U に関して、予測基準 2 2 4 は、同じピクチャ内の（順方向経路内で）符号化され（再構築経路内で）再構築されている 1 つ又は複数の隣接 B P U を含み得る。符号器は、隣接 B P U を外挿することによって、予測済み B P U 2 0 8 を生成することができる。外挿技法は、例えば線形外挿又は線形補間、多項式外挿又は多項式補間等を含み得る。一部の実施形態では、符号器は、予測済み B P U 2 0 8 のピクセルごとに、対応するピクセルの値を外挿することによって等、ピクセルレベルで外挿を行うことができる。外挿に使用される隣接 B P U は、垂直方向（例えば元の B P U の上）、水平方向（例えば元の B P U の左）、対角線方向（例えば元の B P U の左下、右下、左上、又は右上）、又は使用される映像コード化規格内で規定される任意の方向等、様々な方向から元の B P U に対して位置し得る。イントラ予測では、予測データ 2 0 6 は、例えば使用される隣接 B P U の位置（例えば座標）、使用される隣接 B P U のサイズ、外挿のパラメータ、元の B P U に対する使用される隣接 B P U の方向等を含み得る。

【 0 0 4 7 】

[0064] 別の例では、符号器は、時間的予測段階 2 0 4 4 でインタ予測を行うことがで

10

20

30

40

50

きる。現ピクチャの元のBPUに関して、予測基準224は、(順方向経路内で)符号化され(再構築経路内で)再構築されている1つ又は複数のピクチャ(「参照ピクチャ」と呼ぶ)を含み得る。一部の実施形態では、参照ピクチャがBPUごとに符号化され再構築され得る。例えば符号器は、再構築済み残差BPU222を予測済みBPU208に加えて再構築済みBPUを生成することができる。同じピクチャの全ての再構築済みBPUが生成されると、符号器は、参照ピクチャとして再構築済みピクチャを生成することができる。符号器は、参照ピクチャの範囲(「探索窓」と呼ぶ)内の一致領域を探すために「動き推定」の操作を行うことができる。参照ピクチャ内の探索窓の位置は、現ピクチャ内の元のBPUの位置に基づいて決定することができる。例えば探索窓は、参照ピクチャ内で、現ピクチャ内の元のBPUと同じ座標を有する位置に中心を置くことができ、所定の距離にわたって広げることができる。符号器が探索窓内で元のBPUと同様の領域を(例えばpel再帰アルゴリズム、ブロックマッチングアルゴリズム等を使用することによって)識別すると、符号器は、その領域を一致領域として決定することができる。一致領域は元のBPUと異なる(例えばそれよりも小さい、等しい、大きい、又は異なる形状の)寸法を有し得る。参照ピクチャ及び現ピクチャは(例えば図1に示すように)タイムライン内で時間的に隔てられているので、時間が経つにつれて一致領域が元のBPUの位置に「移動する」と見なすことができる。符号器はかかる動きの方向及び距離を「動きベクトル」として記録することができる。(例えば図1のピクチャ106のような)複数の参照ピクチャが使用される場合、符号器は、参照ピクチャごとに一致領域を探し、その関連する動きベクトルを求めることができる。一部の実施形態では、符号器が個々の一致する参照ピクチャの一致領域のピクセル値に重みを割り当てることができる。

10

20

【0048】

[0065] 動き推定は、例えば平行移動、回転、拡大縮小等の様々な種類の動きを識別するために使用することができる。インタ予測では、予測データ206は、例えば一致領域の位置(例えば座標)、一致領域に関連する動きベクトル、参照ピクチャの数、参照ピクチャに関連する重み等を含み得る。

【0049】

[0066] 予測済みBPU208を生成するために、符号器は「動き補償」の操作を行うことができる。動き補償は、予測データ206(例えば動きベクトル)及び予測基準224に基づいて予測済みBPU208を再構築するために使用することができる。例えば符号器は、動きベクトルに従って参照ピクチャの一致領域を動かすことができ、その中で符号器は現ピクチャの元のBPUを予測することができる。(例えば図1のピクチャ106のような)複数の参照ピクチャが使用される場合、符号器は、個々の動きベクトルに従って参照ピクチャの一致領域を動かし、一致領域のピクセル値を平均することができる。一部の実施形態では、符号器が個々の一致する参照ピクチャの一致領域のピクセル値に重みを割り当てた場合、符号器は、動かした一致領域のピクセル値の加重和を求めることができる。

30

【0050】

[0067] 一部の実施形態では、インタ予測は単方向でも双方向でもよい。単方向のインタ予測は、現ピクチャに対して同じ時間的方向にある1つ又は複数の参照ピクチャを使用することができる。例えば図1のピクチャ104は、参照ピクチャ(即ちピクチャ102)がピクチャ104に先行する単方向のインタ予測ピクチャである。双方向のインタ予測は、現ピクチャに対して両方の時間的方向にある1つ又は複数の参照ピクチャを使用することができる。例えば図1のピクチャ106は、参照ピクチャ(即ちピクチャ104及び108)がピクチャ104に対して両方の時間的方向にある双方向のインタ予測ピクチャである。

40

【0051】

[0068] プロセス200Bの順方向経路を引き続き参照し、空間的予測段階2042及び時間的予測段階2044の後、モード決定段階230において、符号器は、プロセス200Bの現在の反復のための予測モード(例えばイントラ予測又はインタ予測の1つ)を

50

選択することができる。例えば符号器はレート歪み最適化技法を実行することができ、かかる技法では、符号器は、候補予測モードのビットレート及び候補予測モード下の再構築済み参照ピクチャの歪みに応じてコスト関数の値を最小化するための予測モードを選択することができる。選択される予測モードに応じて、符号器は、対応する予測済みBPU208及び予測済みデータ206を生成することができる。

【0052】

[0069] プロセス200Bの再構築経路において、順方向経路内でイントラ予測モードが選択されている場合、予測基準224（例えば現ピクチャ内で符号化され再構築されている現BPU）を生成した後で、符号器は、後で使用するために（例えば現ピクチャの次のBPUを外挿するために）空間的予測段階2042に予測基準224を直接フィードすることができる。順方向経路内でインタ予測モードが選択されている場合、予測基準224（例えば全てのBPUが符号化され再構築されている現ピクチャ）を生成した後で、符号器は、ループフィルタ段階232に予測基準224をフィードすることができ、ループフィルタ段階232では、符号器は、予測基準224にループフィルタを適用してインタ予測によって引き起こされる歪み（例えばブロッキングアーティファクト）を減らし又はなくすことができる。例えばデブロッキング、サンプル適応オフセット、適応ループフィルタ等、符号器は、ループフィルタ段階232で様々なループフィルタ技法を適用することができる。ループフィルタ済みの参照ピクチャは、後で使用するために（例えば映像シーケンス202の将来のピクチャ用のインタ予測参照ピクチャとして使用するために）バッファ234（又は「復号済みピクチャバッファ」）内に記憶することができる。符号器は、時間的予測段階2044で使用するために1つ又は複数の参照ピクチャをバッファ234内に記憶することができる。一部の実施形態では、符号器は、量子化済み変換係数216、予測データ206、及び他の情報と共にループフィルタのパラメータ（例えばループフィルタの強度）をバイナリコード化段階226で符号化することができる。

【0053】

[0070] 図3Aは、本開示の一部の実施形態による復号プロセス300Aの一例の概略図を示す。プロセス300Aは、図2Aの圧縮プロセス200Aに対応する解凍プロセスであり得る。一部の実施形態では、プロセス300Aがプロセス200Aの再構築経路と同様であり得る。復号器は、プロセス300Aに従って映像ビットストリーム228を映像ストリーム304へと復号することができる。映像ストリーム304は映像シーケンス202と非常に似ているものであり得る。しかし、圧縮及び解凍プロセス（例えば図2A～図2Bの量子化段階214）における情報損失により、概して映像ストリーム304は映像シーケンス202と同一ではない。図2A～図2Bのプロセス200A及び200Bと同様に、復号器は、映像ビットストリーム228内に符号化される各ピクチャについて基本処理単位（BPU）のレベルにおいてプロセス300Aを実行することができる。例えば復号器はプロセス300Aを反復的なやり方で実行することができ、復号器はプロセス300Aの1回の反復の中で基本処理単位を復号することができる。一部の実施形態では、復号器は、映像ビットストリーム228内に符号化される各ピクチャの領域（例えば領域114～118）についてプロセス300Aを並列に実行することができる。

【0054】

[0071] 図3Aでは、復号器は、符号化済みピクチャの基本処理単位（「符号化済みBPU」と呼ぶ）に関連する映像ビットストリーム228の一部をバイナリ復号段階302にフィードすることができる。バイナリ復号段階302で、復号器は、その部分を予測データ206及び量子化済み変換係数216へと復号することができる。復号器は、量子化済み変換係数216を逆量子化段階218及び逆変換段階220にフィードして、再構築済み残差BPU222を生成することができる。復号器は、予測データ206を予測段階204にフィードして、予測済みBPU208を生成することができる。復号器は、再構築済み残差BPU222を予測済みBPU208に加えて、予測済み基準224を生成することができる。一部の実施形態では、予測済み基準224は、バッファ（例えばコンピュータメモリ内の復号済みピクチャバッファ）内に記憶され得る。復号器は、プロセス3

10

20

30

40

50

00Aの次の反復内で予測操作を行うための予測済み基準224を、予測段階204にフィードすることができる。

【0055】

[0072] 復号器は、プロセス300Aを反復的に実行して、符号化済みピクチャの各符号化済みBPUを復号し、符号化済みピクチャの次の符号化済みBPUを符号化するための予測済み基準224を生成することができる。符号化済みピクチャの全ての符号化済みBPUを復号した後、復号器は、表示するためにピクチャを映像ストリーム304に出力し、映像ビットストリーム228内の次の符号化済みピクチャの復号に進むことができる。

【0056】

[0073] バイナリ復号段階302で、復号器は、符号器が使用したバイナリコード化技法（例えばエントロピコード化、可変長コード化、算術コード化、ハフマンコード化、コンテキスト適応バイナリ算術コード化、又は他の任意の可逆圧縮アルゴリズム）の逆操作を行うことができる。一部の実施形態では、予測データ206及び量子化済み変換係数216に加えて、復号器は、例えば予測モード、予測操作のパラメータ、変換の種類、量子化プロセスのパラメータ（例えば量子化パラメータ）、符号器制御パラメータ（例えばビットレート制御パラメータ）等の他の情報を、バイナリ復号段階302において復号することができる。一部の実施形態では、映像ビットストリーム228がネットワーク上でパケット単位で伝送される場合、復号器は、バイナリ復号段階302にフィードする前に、映像ビットストリーム228をパケット化解除することができる。

【0057】

[0074] 図3Bは、本開示の一部の実施形態による復号プロセスの別の例300Bの概略図を示す。プロセス300Bはプロセス300Aから修正され得る。例えばプロセス300Bは、ハイブリッド映像コード化規格（例えばH.26xシリーズ）に準拠する復号器によって使用され得る。プロセス300Aと比較し、プロセス300Bは、予測段階204を空間的予測段階2042及び時間的予測段階2044に更に分け、ループフィルタ段階232及びバッファ234を追加で含む。

【0058】

[0075] プロセス300Bでは、復号されている符号化済みピクチャ（「現ピクチャ」と呼ぶ）の符号化済み基本処理単位（「現BPU」と呼ぶ）に関して、復号器によってバイナリ復号段階302から復号される予測データ206は、現BPUを符号化するためにどの予測モードが符号器によって使用されたのかに応じて様々な種類のデータを含み得る。例えば現BPUを符号化するためにイントラ予測が符号器によって使用された場合、予測データ206は、イントラ予測、イントラ予測操作のパラメータ等を示す予測モードインジケータ（例えばフラグ値）を含み得る。イントラ予測操作のパラメータは、例えば基準として使用される1つ又は複数の隣接BPUの位置（例えば座標）、隣接BPUのサイズ、外挿のパラメータ、元のBPUに対する隣接BPUの方向等を含み得る。別の例では、現BPUを符号化するためにインタ予測が符号器によって使用された場合、予測データ206は、インタ予測、インタ予測操作のパラメータ等を示す予測モードインジケータ（例えばフラグ値）を含み得る。インタ予測操作のパラメータは、例えば現BPUに関連する参照ピクチャの数、参照ピクチャにそれぞれ関連する重み、それぞれの参照ピクチャ内の1つ又は複数の一致領域の位置（例えば座標）、一致領域にそれぞれ関連する1つ又は複数の動きベクトル等を含み得る。

【0059】

[0076] 予測モードインジケータに基づき、復号器は、空間的予測段階2042で空間的予測（例えばイントラ予測）を行うのか、時間的予測段階2044で時間的予測（例えばインタ予測）を行うのかを決めることができる。かかる空間的予測又は時間的予測の実行の詳細は図2Bに示しており、以下で繰り返さない。かかる空間的予測又は時間的予測を行った後、復号器は、予測済みBPU208を生成することができる。図3Aに記載したように、復号器は、予測済みBPU208と再構築済み残差BPU222とを加えて、予測基準224を生成することができる。

10

20

30

40

50

【 0 0 6 0 】

[0077] プロセス 3 0 0 B では、復号器は、プロセス 3 0 0 B の次の反復内で予測操作を行うための予測済み基準 2 2 4 を、空間的予測段階 2 0 4 2 又は時間的予測段階 2 0 4 4 にフィードすることができる。例えば現 B P U が空間的予測段階 2 0 4 2 においてインタ予測を使用して復号される場合、予測基準 2 2 4 (例えば復号済みの現 B P U) を生成した後で、復号器は、後で使用するために (例えば現ピクチャの次の B P U を外挿するために) 空間的予測段階 2 0 4 2 に予測基準 2 2 4 を直接フィードすることができる。現 B P U が時間的予測段階 2 0 4 4 においてインタ予測を使用して復号される場合、予測基準 2 2 4 (例えば全ての B P U が復号されている参照ピクチャ) を生成した後で、符号器は、ループフィルタ段階 2 3 2 に予測基準 2 2 4 をフィードして、歪み (例えばブロッキングアーティファクト) を減らし又はなくすことができる。復号器は、図 2 B に記載したやり方で予測基準 2 2 4 にループフィルタを適用することができる。ループフィルタ済みの参照ピクチャは、後で使用するために (例えば映像ビットストリーム 2 2 8 の将来の符号化ピクチャ用のインタ予測参照ピクチャとして使用するために) バッファ 2 3 4 (例えばコンピュータメモリ内の復号済みピクチャバッファ) 内に記憶することができる。復号器は、時間的予測段階 2 0 4 4 で使用するために 1 つ又は複数の参照ピクチャをバッファ 2 3 4 内に記憶することができる。一部の実施形態では、現 B P U を符号化するためにインタ予測が使用されたことを予測データ 2 0 6 の予測モードインジケータが示す場合、予測データはループフィルタのパラメータ (例えばループフィルタの強度) を更に含むことができる。

10

20

【 0 0 6 1 】

[0078] 図 4 は、本開示の一部の実施形態による、映像を符号化又は復号するための機器 4 0 0 の一例のブロック図である。図 4 に示すように、機器 4 0 0 はプロセッサ 4 0 2 を含み得る。プロセッサ 4 0 2 が本明細書に記載の命令を実行するとき、機器 4 0 0 は映像を符号化し又は復号するための専用マシンになり得る。プロセッサ 4 0 2 は情報を操作し又は処理することができる任意の種類の回路であり得る。例えばプロセッサ 4 0 2 は、任意の数の中央処理装置 (「CPU」)、グラフィックス処理装置 (「GPU」)、ニューラル処理ユニット (「NPU」)、マイクロコントローラユニット (「MCU」)、光プロセッサ、プログラム可能論理コントローラ、マイクロコントローラ、マイクロプロセッサ、デジタル信号プロセッサ、IP (intellectual property) コア、プログラム可能論理アレイ (PLA)、プログラム可能アレイ論理 (PAL)、汎用アレイ論理 (GAL)、複合プログラム可能論理装置 (CPLD)、書換可能ゲートアレイ (FPGA)、システムオンチップ (SOC)、特定用途向け集積回路 (ASIC) 等の任意の組み合わせを含み得る。一部の実施形態では、プロセッサ 4 0 2 は単一の論理構成要素としてグループ化される 1 組のプロセッサとすることもできる。例えば図 4 に示すように、プロセッサ 4 0 2 はプロセッサ 4 0 2 a、プロセッサ 4 0 2 b、及びプロセッサ 4 0 2 n を含む複数のプロセッサを含み得る。

30

【 0 0 6 2 】

[0079] 機器 4 0 0 は、データ (例えば 1 組の命令、コンピュータコード、中間データ等) を記憶するように構成されるメモリ 4 0 4 も含み得る。例えば図 4 に示すように、記憶データは、プログラム命令 (例えばプロセス 2 0 0 A、2 0 0 B、3 0 0 A、又は 3 0 0 B 内の段階を実装するためのプログラム命令) 及び処理用データ (例えば映像シーケンス 2 0 2、映像ビットストリーム 2 2 8、又は映像ストリーム 3 0 4) を含み得る。プロセッサ 4 0 2 は、プログラム命令及び処理用データに (例えばバス 4 1 0 を介して) アクセスし、プログラム命令を実行して処理用データに対する操作又は処理を行うことができる。メモリ 4 0 4 は、高速ランダムアクセス記憶装置又は不揮発性記憶装置を含み得る。一部の実施形態では、メモリ 4 0 4 が、任意の数のランダムアクセスメモリ (RAM)、読取専用メモリ (ROM)、光学ディスク、磁気ディスク、ハードドライブ、ソリッドステートドライブ、フラッシュドライブ、セキュリティデジタル (SD) カード、メモリスティック、コンパクトフラッシュ (登録商標) (CF) カード等の任意の組み合わせを含

40

50

み得る。メモリ 404 は、単一の論理構成要素としてグループ化される（図 4 には不図示の）メモリ群とすることもできる。

【0063】

[0080] 内蔵バス（例えば CPU メモリバス）、外部バス（例えばユニバーサルシリアルバスポート、周辺機器コンポーネント相互接続エクスプレスポート）等のバス 410 は、機器 400 内の構成要素間でデータを転送する通信装置であり得る。

【0064】

[0081] 曖昧さを招くことなしに説明を簡単にするために、本開示ではプロセッサ 402 及び他のデータ処理回路をまとめて「データ処理回路」と呼ぶ。データ処理回路は、完全にハードウェアとして、又はソフトウェア、ハードウェア、若しくはファームウェアの組み合わせとして実装することができる。加えてデータ処理回路は、単一の独立したモジュールとすることができ、又は機器 400 の他の任意の構成要素内に完全に若しくは部分的に組み合わせることができる。

10

【0065】

[0082] 機器 400 は、ネットワーク（例えばインターネット、イントラネット、ローカルエリアネットワーク、モバイル通信ネットワーク等）との有線通信又は無線通信を提供するためのネットワークインタフェース 406 を更に含み得る。一部の実施形態では、ネットワークインタフェース 406 が任意の数のネットワークインタフェースコントローラ（NIC）、無線周波数（RF）モジュール、トランスポンダ、トランシーバ、モデム、ルータ、ゲートウェイ、有線ネットワークアダプタ、無線ネットワークアダプタ、Bluetooth アダプタ、赤外線アダプタ、近距離無線通信（「NFC」）アダプタ、セルラネットワークチップ等の任意の組み合わせを含み得る。

20

【0066】

[0083] 一部の実施形態では、機器 400 は、1 つ又は複数の周辺装置への接続を提供するための周辺装置インタフェース 408 を任意選択的に更に含み得る。図 4 に示すように、周辺装置はこれだけに限定されないが、カーソル制御装置（例えばマウス、タッチパッド、又はタッチスクリーン）、キーボード、ディスプレイ（例えばブラウン管ディスプレイ、液晶ディスプレイ、又は発光ダイオードディスプレイ）、映像入力装置（例えば映像アーカイブに結合されるカメラ又は入力インタフェース）等を含み得る。

【0067】

30

[0084] 映像コーデック（例えばプロセス 200A、200B、300A、又は 300B を実行するコーデック）は、機器 400 内の任意のソフトウェア又はハードウェアモジュールの任意の組み合わせとして実装できることに留意すべきである。例えばプロセス 200A、200B、300A、又は 300B の一部の又は全ての段階が、メモリ 404 内にロード可能なプログラム命令等の機器 400 の 1 つ又は複数のソフトウェアモジュールとして実装され得る。別の例では、プロセス 200A、200B、300A、又は 300B の一部の又は全ての段階が、専用データ処理回路（例えば FPG A、ASIC、NPU 等）等の機器 400 の 1 つ又は複数のハードウェアモジュールとして実装され得る。

【0068】

[0085] 図 5 は、本開示の一部の実施形態による、例示的なクロマスケーリングによるルママッピング（LMCS）プロセス 500 の概略図を示す。例えばプロセス 500 は、ハイブリッド映像コード化規格（例えば H.26x シリーズ）に準拠する復号器によって使用され得る。LMCS は、図 2B のループフィルタ 232 の前に適用される新たな処理ブロックである。LMCS はリシェーパと呼ぶこともできる。

40

【0069】

[0086] LMCS プロセス 500 は、適応区分線形モデルに基づくルマ成分値のインルーブマッピング及びクロマ成分のルマ依存クロマ残差スケーリングを含み得る。

【0070】

[0087] 図 5 に示すように、適応区分線形モデルに基づくルマ成分値のインルーブマッピングは、順方向マッピング段階 518 及び逆マッピング段階 508 を含み得る。クロマ

50

成分のルマ依存クロマ残差スケーリングは、クロマスケーリング 5 2 0 を含み得る。

【 0 0 7 1 】

[0088] マッピング前の又は逆マッピング後のサンプル値は元の領域内のサンプルと呼ぶことができ、マッピング後の及び逆マッピング前のサンプル値はマップ領域内のサンプルと呼ぶことができる。LMCS が有効化される場合、プロセス 5 0 0 内の一部の段階は元の領域ではなくマップ領域内で行うことができる。順方向マッピング段階 5 1 8 及び逆マッピング段階 5 0 8 は、SPS フラグを使用してシーケンスレベルで有効化 / 無効化で
 10

【 0 0 7 2 】

[0089] 図 5 に示すように、 Q^{-1} & T^{-1} 段階 5 0 4、再構築 5 0 6、及びイントラ
 予測 5 0 8 はマップ領域内で実行される。例えば Q^{-1} & T^{-1} 段階 5 0 4 は逆量子化及び
 逆変換を含むことができ、再構築 5 0 6 はルマ予測とルマ残差との加算を含むことができ
 、イントラ予測 5 0 8 はルマイントラ予測を含み得る。

【 0 0 7 3 】

[0090] ループフィルタ 5 1 0、動き補償段階 5 1 6 及び 5 3 0、イントラ予測段階 5
 2 8、再構築段階 5 2 2、並びに復号済みピクチャバッファ (DPB) 5 1 2 及び 5 2 6
 は元の (即ち非マップ) 領域内で実行される。一部の実施形態では、ループフィルタ 5 1
 0 がデブロッキング、適応ループフィルタ (ALF)、及びサンプル適応オフセット (S
 AO) を含むことができ、再構築段階 5 2 2 がクロマ予測とクロマ残差との加算を含むこ
 とができ、DPB 5 1 2 及び 5 2 6 が復号済みピクチャを参照ピクチャとして記憶するこ
 とができる。
 20

【 0 0 7 4 】

[0091] 一部の実施形態では、区分線形モデルによるルママッピングを適用することが
 できる。

【 0 0 7 5 】

[0092] ルマ成分のインループマッピングは、ダイナミックレンジにわたるコード語の
 再配分によって入力映像の信号統計を調節して圧縮効率を改善することができる。ルママ
 ッピングは、順方向マッピング関数「FwdMap」及び対応する逆マッピング関数「InvMa
 p」によって実行することができる。「FwdMap」関数は 1 6 個の等しい区分と共に区分
 線形モデルを使用してシグナリングされる。「InvMap」関数はシグナリングされる必要
 がなく、代わりに「FwdMap」関数から導出される。
 30

【 0 0 7 6 】

[0093] 区分線形モデルのシグナリングを図 6 の表 1 及び図 7 の表 2 に示す。図 6 の表
 1 は、タイルグループヘッダ構文構造を示す。図 6 に示すように、現タイルグループ内に
 ルママッピングモデルがあるかどうかを示すためにリシェーパモデルパラメータ存在フラ
 グがシグナリングされる。現タイルグループ内にルママッピングモデルがある場合、図 7
 の表 2 に示す構文要素を使用し、対応する区分線形モデルパラメータをtile_group_resh
 aper_model()内でシグナリングすることができる。区分線形モデルは入力信号のダイナ
 ミックレンジを 1 6 個の等しい区分に分割する。1 6 個の等しい区分のそれぞれについて
 、区分に割り当てられるコード語の数を使用して区分の線形マッピングパラメータが表さ
 れる。1 0 ビット入力を例に取る。1 6 個の区分のそれぞれは、その区分に割り当てられ
 る 6 4 のコード語をデフォルトで有し得る。スケール因子を計算し、その区分についてマ
 ッピング関数をしかるべく調節するために、シグナリングされるコード語の数を使用す
 ることができる。図 7 の表 2 は、コード語の数がシグナリングされる最小インデックス「res
 haper_model_min_bin_idx」及び最大インデックス「reshaper_model_max_bin_idx
 」も包含的に定める。区分インデックスがreshaper_model_min_bin_idxよりも小さい
 又はreshaper_model_max_bin_idxよりも大きい場合、その区分のためのコード語の数は
 シグナリングされず、ゼロだと推論される (即ちその区分にコード語が割り当てられず
 、マッピング / スケーリングが適用されない)。
 40

【 0 0 7 7 】

【0094】 tile_group_reshaper_model()がシグナリングされた後、図8に示すLMCSプロセスが現タイルグループに適用されるかどうかを示すために、タイルグループヘッダレベルで別のリシェーパイネーブルフラグ「tile_group_reshaper_enable_flag」がシグナリングされる。リシェーパが現タイルグループについて有効化され、現タイルグループがデュアルツリー分割を使用しない場合、クロマスケーリングが現タイルグループについて有効化されるかどうかを示すために、更なるクロマスケーリングイネーブルフラグがシグナリングされる。デュアルツリー分割はクロマセパレートツリーとも呼ぶことができる。

【0078】

【0095】 区分線形モデルは、図7の表2内のシグナリングされる構文要素に基づいて以下のように構築することができる。「FwdMap」区分線形モデルのそれぞれの*i*番目の区分、 $i = 0, 1, \dots, 15$ は、2つの入力ピボットポイントInputPivot[]及び2つの出力（マップ済み）ピボットポイントMappedPivot[]によって定められる。InputPivot[]及びMappedPivot[]はシグナリングされる構文に基づいて以下のように計算される（一般性を失うことなく入力映像のビット深度を10ビットだと仮定する）：

1) OrgCW=64

2) $i=0:16$ ではInputPivot[i]= $i \cdot \text{OrgCW}$

3) $i=\text{reshaper_model_min_bin_idx} : \text{reshaper_model_max_bin_idx}$ ではSignaledCW[i]= $\text{OrgCW} + (1 - 2 \cdot \text{reshape_model_bin_delta_sign_CW}[i]) \cdot \text{reshape_model_bin_delta_abs_CW}[i]$;

4) $i=0:16$ ではMappedPivot[i]を以下のように計算する：

MappedPivot[0]=0;

($i=0; i < 16; i++$)では

MappedPivot[i+1]=MappedPivot[i]+SignaledCW[i]

【0079】

【0096】 逆マッピング関数「InvMap」もInputPivot[]及びMappedPivot[]によって定めることができる。「FwdMap」と異なり、「InvMap」区分線形モデルでは、各区分の2つの入力ピボットポイントをMappedPivot[]によって定めることができ、2つの出力ピボットポイントをInputPivot[]によって定めることができ、これは「FwdMap」の反対である。このようにして「FwdMap」の入力は等しい区分に分割されるが、「InvMap」の入力が等しい区分に分割されることは保証されない。

【0080】

【0097】 図5に示すように、インタコード化ブロックでは動き補償予測をマップ領域内で実行することができる。換言すれば、動き補償予測516の後、DPB内内の参照信号に基づいて Y_{pred} を計算し、元の領域内のルマ予測ブロックをマップ領域にマップするために「FwdMap」関数518を適用することができる（ $Y_{pred} = \text{FwdMap}(Y_{pred})$ ）。インタコード化されたブロックでは、イントラ予測内で使用された参照サンプルがマップ領域内に既にあるので「FwdMap」関数は適用されない。再構築済みブロック506の後、 Y_r を計算することができる。マップ領域内の再構築済みルマ値を元の領域内の再構築済みルマ値に変換して戻すために「InvMap」関数508を適用することができる（

【数1】

$$\hat{Y}_i = \text{InvMap}(Y_r)$$

）。「InvMap」関数508はイントラコード化されたルマブロック及びインタコード化されたルマブロックの両方に適用することができる。

【0081】

【0098】 ルママッピングプロセス（順方向又は逆マッピング）は、参照表（LUT）を使用して又はその場での計算を使用して実装することができる。LUTが使用される場合、タイルグループレベルでの使用のために「FwdMapLUT[]」及び「InvMapLUT[]」の

41, 2048, 2048, 2048, 1820, 1820, 1820, 1638, 1638, 1638, 1638, 1489, 1489, 1489, 1489, 1365, 1365, 1365, 1365, 1260, 1260, 1260, 1260, 1170, 1170, 1170, 1092, 1092, 1092, 1092, 1024, 1024, 1024, 1024};
shiftC=11

- (SignaledCW[i]==0)が成立する場合、
cScaleInv[i]=(1 shiftC)

- さもないければ

cScaleInv[i]=ChromaResidualScaleLut[(SignaledCW[i] 1)-1]

【 0 0 8 8 】

[0105] 静的テーブルChromaResidualScaleLut[]は64個のエントリを含み、SignaledCW[]は[0,128]の範囲内にある(入力を10ビットと仮定する)。従って、クロマスケール因子LUTのcScaleInv[]を構築するために2による除算(例えば1による右シフト)を使用する。クロマスケール因子LUTのcScaleInv[]は複数のクロマスケール因子を含むことができる。LUTのcScaleInv[]はタイルグループレベルで構築される。

【 0 0 8 9 】

[0106] 現ブロックがイントラ、C I I P、又はイントラブロックコピー(I B C、現ピクチャ参照又はC P Rとしても知られる)モードを使用してコード化される場合、avgY'はイントラ、C I I P、又はI B C予測ルマ値の平均として計算される。さもないければ、avgY'は順方向マップされるインタ予測ルマ値(つまり図5の Y_{pred})の平均として計算される。サンプルに基づいて実行されるルママッピングと異なり、C_{ScaleInv}は全クロマブロックの定数値である。C_{ScaleInv}を用いてクロマ残差スケーリングを以下のように復号器側で適用する：

【数3】

$$\text{Decoder side: } \hat{C}_{Res} = \hat{C}_{ResScale} * C_{ScaleInv}$$

【 0 0 9 0 】

[0107] 但し

【数4】

$$\hat{C}_{ResScale}$$

は現ブロックの再構築済みクロマ残差である。符号器側で、順方向クロマ残差スケーリング(変換され量子化される前)を次のように行う： $C_{ResScale} = C_{Res} * C_{Scale} = C_{Res} / C_{ScaleInv}$ 。

【 0 0 9 1 】

[0108] 一部の実施形態ではデュアルツリー分割が行われ得る。

【 0 0 9 2 】

[0109] V V Cのドラフト4では、コード化ツリー方式は、ルマ及びクロマが別個のブロックツリー分割を有する能力をサポートする。これはデュアルツリー分割とも呼ばれる。デュアルツリー分割のシグナリングを図8の表3及び図9の表4に示す。S P S内でシグナリングされるシーケンスレベル制御フラグである「qtbtt_dual_tree_intra_flag」がオンにされ、現タイルグループがイントラコード化される場合、ブロック分割情報をまずルマについて、次いでクロマについて別々にシグナリングすることができる。インタコード化されるタイルグループ(P及びBタイルグループ)ではデュアルツリー分割が許可されない。別個のブロックツリーモードが適用される場合、図9の表4に示すようにルマコード化ツリーブロック(C T B)が或るコード化ツリー構造によってC Uへと分割され、クロマC T Bが別のコード化ツリー構造によってクロマC Uへと分割される。

【 0 0 9 3 】

[0110] ルマ及びクロマが異なる分割を有することが許可される場合、様々な色成分間

の依存性を有するコード化ツールに関して問題が生じ得る。例えばLMCSの場合、現ブロックに適用されるスケール因子を見出すために、対応するルマブロックの平均値を使用する。デュアルツリーが使用される場合、そのことが全CTUのレイテンシを招き得る。例えばCTUのルマブロックが垂直に1回分割され、CTUのクロマブロックが水平に1回分割される場合、CTUの第1のクロマブロックが復号可能である前にCTUのルマブロックの両方が復号される（それによりクロマスケール因子を計算するのに必要な平均値を計算することができる）。VV Cでは、CTUがルマサンプルの単位で 128×128 もの大きさであり得る。そのような大きいレイテンシはハードウェア復号器のパイプライン設計にとって非常に問題となる可能性がある。従って、VV Cのドラフト4は、デュアルツリー分割とルマ依存クロマスケールとの組み合わせを禁止することができる。現
10
タイルグループについてデュアルツリー分割が有効化される場合、クロマスケールを強制的にオフにすることができる。LMCSのルママッピング部分は、ルマ成分に対してのみ影響し、色成分を横断する依存性の問題を有さないので、デュアルツリーの事例でも引き続き許可されていることに留意されたい。より優れたコード化効率を実現するために色成分間の依存性に依拠するコード化ツールの別の例は、クロスコンポーネント線形モデル（CCLM）と呼ばれる。

【0094】

[0111] 従って、タイルグループレベルのクロマスケール因子LUTのcScaleInv[]の導出は容易に拡張することができない。導出プロセスは、64個のエントリを有する一定のクロマLUTChromaResidualScaleLutに現在は依存する。16個の区分を有する1
20
0ビットの映像では、2による除算の追加のステップを適用しなければならない。区分の数が変わる場合、例えば16個の区分の代わりに8個の区分が使用される場合、2による除算ではなく4による除算を適用するように導出プロセスを変更しなければならない。この追加のステップは精度の損失を引き起こすだけでなく、洗練されておらず不要である。

【0095】

[0112] 更に、クロマスケール因子を得るために使用される現クロマブロックの区分インデックスY_{idx}を計算するために、全ルマブロックの平均値を使用することができる。これは望ましくなく大概不要である。 128×128 の最大CTUサイズを検討されたい。この場合、平均ルマ値が16384（ 128×128 ）ルマサンプルに基づいて計算され、かかる計算は複雑である。更に、 128×128 のルマブロック分割が符号器によって
30
選択される場合、そのブロックは均質のコンテンツを含む可能性が高い。従って、ルマ平均を計算するのにブロック内のルマサンプルのサブセットで十分であり得る。

【0096】

[0113] デュアルツリー分割では、ハードウェア復号器の潜在的なパイプラインの問題を回避するためにクロマスケールがオフであり得る。しかし、適用されるクロマスケール因子を導出するために対応するルマサンプルを使用する代わりに、適用されるクロマスケール因子を示すための明確なシグナリングを使用する場合は、この依存性を回避することができる。イントラコード化されるタイルグループ内のクロマスケールを有効化することはコード化効率を更に改善し得る。

【0097】

[0114] 区分線形パラメータのシグナリングは更に改善することができる。現在、16個の区分のそれぞれについてデルタコード語値がシグナリングされる。往々にしてこの16区分について、限られた数の異なるコード語しか使用されないことが認められている。従ってシグナリングのオーバーヘッドを更に減らすことができる。

【0098】

[0115] 本開示の実施形態は、クロマスケールLUTを除去することによって映像コンテンツを処理するための方法を提供する。

【0099】

[0116] 上記で述べたように、64エントリのクロマLUTの拡張は困難な場合があり、他の区分線形モデルが使用される場合（例えば8区分、4区分、64区分等）は問題と
50

なり得る。同じコード化効率を実現するために、クロマスケール因子をその対応する区分のルマスケール因子と同じに設定することができるので、かかる拡張は不要でもある。本開示の一部の実施形態では、クロマスケール因子 (chroma_scaling) を以下のように現クロマブロックの区分インデックス「Y_{idx}」に基づいて決定することができる。

- ・ Y_{idx} reshaper_model_max_bin_idx、Y_{idx} reshaper_model_min_bin_idx又はSignaledCW[Y_{idx}]=0が成立する場合、chroma_scalingをデフォルトに設定しchroma_scaling=1.0となる。

- ・ さもなければchroma_scalingをSignaledCW[Y_{idx}]/OrgCWに設定する。

【 0 1 0 0 】

[0117] chroma_scaling=1.0である場合スケーリングは適用されない。

10

【 0 1 0 1 】

[0118] 上記で決定したクロマスケール因子は小数精度を有し得る。ハードウェア/ソフトウェアプラットフォーム上の依存性を回避するために固定小数点近似を適用できることが理解されよう。更に、復号器側で逆クロマスケールリングを行うことができる。従って、乗算とその後に続く右シフトを使用する固定小数点演算によって除算を実装することができる。固定小数点精度における逆クロマスケール因子「inverse_chroma_scaling[]」は、固定小数点近似「CSCALE_FP_PREC」内のビットの数に基づいて以下のように決定することができる。

$$\text{inverse_chroma_scaling}[Y_{idx}] = ((1 \ll (\text{luma_bit_depth} - \log_2(\text{TOTAL_NUMBER_PIECES}) + \text{CSCALE_FP_PREC})) + (\text{SignaledCW}[Y_{idx}] - 1)) / \text{SignaledCW}[Y_{idx}]$$

20

但しluma_bit_depthはルマビット深度であり、TOTAL_NUMBER_PIECESはVVCのドラフト4では16に設定される区分線形モデル内の総区分数である。「inverse_chroma_scaling[]」の値はタイルグループごとに1回計算するだけでよい場合があり、上記の除算は整数除算演算であることが理解されよう。

【 0 1 0 2 】

[0119] クロマスケール因子及び逆スケール因子を決定するために更なる量子化を適用することができる。例えば「SignaledCW」の全ての偶数 (2 × m) 値について逆クロマスケール因子を計算することができ、「SignaledCW」の奇数 (2 × m + 1) 値は隣接する偶数値のスケール因子のクロマスケール因子を再利用する。換言すれば下記を使用することができる：

30

```
for(i=reshaper_model_min_bin_idx; i ≤ reshaper_model_max_bin_idx; i++)
{
    tempCW=SignaledCW[i] - 1;
    inverse_chroma_scaling[i]=((1 << (luma_bit_depth-log2(TOTAL_NUMBER_PIECES)+CSCALE_FP_PREC))+(tempCW - 1))/tempCW;
}
```

【 0 1 0 3 】

[0120] クロマスケール因子の量子化は更に汎用化することができる。例えば、他の全ての隣接値が同じクロマスケール因子を共有しながら「SignaledCW」のn番目の値ごとに逆クロマスケール因子「inverse_chroma_scaling[]」を計算することができる。例えば「n」は4に設定することができる。従って、隣接する4個のコード語値ごとに同じ逆クロマスケール因子値が共有され得る。一部の実施形態では「n」の値を2のべき乗とすることができ、かかる設定は除算を計算するためにシフトを使用することを可能にする。log₂(n)の値をLOG2_nとして表し、上記の等式「tempCW=SignaledCW[i] - 1」は以下のように修正することができる：

40

$$\text{tempCW} = \text{SignaledCW}[i] \ll \text{LOG2_n} - \text{LOG2_n}$$

【 0 1 0 4 】

[0121] 一部の実施形態では、LOG2_nの値は区分線形モデル内で使用される区分数の関数であり得る。より少ない区分が使用される場合、より大きいLOG2_nを使用することが有益であり得る。例えばTOTAL_NUMBER_PIECESが16以下の場合、LOG2_nは1+(

50

4- $\log_2(\text{TOTAL_NUMBER_PIECES})$)に設定することができる。TOTAL_NUMBER_PIECESが16を上回る場合、LOG2_nは0に設定することができる。

【0105】

[0122] 本開示の実施形態は、ルマ予測ブロックの平均化を単純化することによって映像コンテンツを処理するための方法を提供する。

【0106】

[0123] 上記で論じたように、現クロマブロックの区分インデックス「Y_{idx}」を決定するために、対応するルマブロックの平均値を使用することができる。しかし、大きいブロックのサイズでは平均化プロセスが多数のルマサンプルを含み得る。最悪の場合、128×128のルマサンプルが平均化プロセスに関与し得る。

【0107】

[0124] 本開示の実施形態は、最悪の場合を僅かN×Nのルマサンプル（Nは2のべき乗である）を使用することに低減するための単純化した平均化プロセスを提供する。

【0108】

[0125] 一部の実施形態では、2次元ルマブロックの両方の寸法が事前設定閾値M以下である（換言すれば2つの寸法の少なくとも一方がMを上回る）場合、その寸法内のMの位置だけを使用するように「ダウンサンプリング」を適用することができる。一般性を失うことなしに横寸法を例に取る。幅がMを上回る場合、位置x（ $x=i \times (\text{width} \div \log_2(M))$, $i=0, \dots, M-1$ ）にあるサンプルだけを平均化に使用する。

【0109】

[0126] 図10は、16×8のルマブロックの平均を計算するために、提案した単純化を適用する一例を示す。この例ではMが4に設定され、ブロック内の16個のルマサンプル（陰影付きのサンプル）だけが平均化に使用される。事前設定閾値Mは4に限定されず、Mは2のべき乗である任意の値に設定できることが理解されよう。例えば事前設定閾値Mは1、2、4、8等であり得る。

【0110】

[0127] 一部の実施形態では、ルマブロックの横寸法及び縦寸法が異なる事前設定閾値Mを有し得る。換言すれば、平均化演算の最悪の場合はM₁×M₂のサンプルを使用し得る。

【0111】

[0128] 一部の実施形態では、寸法を考慮することなしに平均化プロセス内でサンプル数を限定することができる。例えば最大で16個のサンプルを使用することができ、それらのサンプルは1×16、16×1、2×8、8×2、又は4×4形式で横寸法又は縦寸法内に分散することができ、現ブロックの形状に適合する如何なる形式も選択することができる。例えばブロックが縦長の場合は2×8サンプルの行列を使用することができ、ブロックが横長の場合は8×2サンプルの行列を使用することができ、ブロックが正方形の場合は4×4サンプルの行列を使用することができる。

【0112】

[0129] 大きいブロックのサイズが選択される場合、ブロック内のコンテンツがより均質になる傾向があることが理解されよう。従って上記の単純化は平均値と全ルマブロックの真の平均との間の差異を引き起こし得るが、かかる差異は小さいものであり得る。

【0113】

[0130] 更に、復号器側動きベクトルリファインメント（DMVR）は、動き補償が適用可能である前に、復号器が動き検出を行って動きベクトルを導出することを要求する。従って、とりわけ復号器にとってDMVRモードはVVC規格内で複雑であり得る。双方向オプティカルフロー（BDOF）はルマ予測ブロックを得るためにDMVR後に適用しなければならない追加の逐次プロセスなので、VVC規格内のBDOFモードはこの状況を更に複雑にし得る。クロマスケーリングは対応するルマ予測ブロックの平均値を必要とするので、平均値が計算可能である前にDMVR及びBDOFを適用することができる。

【0114】

10

20

30

40

50

【0131】 このレイテンシの問題を解決するために、本開示の一部の実施形態ではDMVR及びBD OFの前にルマ予測ブロックを使用して平均ルマ値を計算し、平均ルマ値を使用してクロマスケール因子を得る。このことはDMVR及びBD OFプロセスと並列にクロマスケールリングを適用できるようにし、従ってレイテンシを著しく減らすことができる。

【0115】

【0132】 本開示と合致して、レイテンシ低減の改変形態を考えることができる。一部の実施形態では、このレイテンシ低減は平均ルマ値を計算するためにルマ予測ブロックの一部しか使用しない上記の単純化した平均化プロセスと組み合わせることもできる。一部の実施形態では、ルマ予測ブロックをDMVRプロセスの後及びBD OFプロセスの前に使用して平均ルマ値を計算することができる。次いで、平均ルマ値を使用してクロマスケール因子を得る。この設計は、クロマスケール因子を決定する精度を保ちながら、BD OFプロセスと並列にクロマスケールリングを適用できるようにする。DMVRプロセスは動きベクトルをリファインする場合があるので、DMVRプロセス前の動きベクトルと共に予測サンプルを使用することよりも、DMVRプロセス後のリファインされた動きベクトルと共に予測サンプルを使用する方が正確であり得る。

10

【0116】

【0133】 更に、VVC規格ではCU構文構造「coding_unit()」が、現CU内に何らかの非ゼロ残差係数があるかどうかを示すための構文要素「cu_cbf」を含む。TUレベルでは、TU構文構造「transform_unit()」が、現TU内に何らかの非ゼロクロマ(Cb又はCr)残差係数があるかどうかを示すための構文要素「tu_cbf_cb」及び「tu_cbf_cr」を含む。従来、VVCのドラフト4では、タイルグループレベルでクロマスケールリングが有効化される場合は、対応するルマブロックの平均化が常に呼び出される。

20

【0117】

【0134】 本開示の実施形態は、ルマ平均化プロセスをバイパスすることによって映像コンテンツを処理するための方法を更に提供する。開示する実施形態と合致して、クロマスケールリングプロセスは残差クロマ係数に適用されるので、非ゼロクロマ係数がない場合はルマ平均化プロセスをバイパスすることができる。これは以下の条件に基づいて決定することができる：

条件1：cu_cbfが0に等しい

条件2：tu_cbf_cr及びtu_cbf_cbがどちらも0に等しい

30

【0118】

【0135】 上記で論じたように「cu_cbf」は現CU内に何らかの非ゼロ残差係数があるかどうかを示すことができ、「tu_cbf_cb」及び「tu_cbf_cr」は現TU内に何らかの非ゼロクロマ(Cb又はCr)残差係数があるかどうかを示すことができる。条件1又は条件2が満たされる場合はルマ平均化プロセスをバイパスすることができる。

【0119】

【0136】 一部の実施形態では、予測ブロックのN×Nサンプルだけを使用して平均値を導出し、そのことは平均化プロセスを単純化する。例えばNが1に等しい場合、予測ブロックの左上サンプルだけが使用される。但し、予測ブロックを使用するこの単純化した平均化プロセスは予測ブロックの生成を依然として必要とし、それによりレイテンシを引き起こす。

40

【0120】

【0137】 一部の実施形態では、クロマスケール因子を生成するために参照ルマサンプルを直接使用することができる。このことは復号器がルマ予測プロセスと並列にスケール因子を導出できるようにし、それによりレイテンシを低減する。以下、参照ルマサンプルを用いたイントラ予測及びインタ予測を別々に説明する。

【0121】

【0138】 例示的なイントラ予測では、予測ブロックを生成するための参照サンプルとして同じピクチャ内の復号済み隣接サンプルを使用することができる。これらの参照サンプルは、例えば現ブロックの上部のサンプル、現ブロックの左側のサンプル、又は現ブロッ

50

クの左上のサンプルを含み得る。これらの参照サンプルの平均を使用してクロマスケール因子を導出することができる。一部の実施形態では、これらの参照サンプルの一部の平均を使用することができる。例えば現ブロックの左上の位置に最も近いK個の参照サンプル（例えばK = 3）だけを平均化する。

【0122】

[0139] 例示的なインタ予測では、予測ブロックを生成するために時間的参照ピクチャからの参照サンプルを使用することができる。これらの参照サンプルは、参照ピクチャインデックス及び動きベクトルによって識別される。動きベクトルが小数精度を有する場合は補間を適用することができる。参照サンプルの平均を決定するために使用する参照サンプルは、補間前又は補間後の参照サンプルを含み得る。補間前の参照サンプルは整数精度にクリッピングされる動きベクトルを含み得る。開示する実施形態と合致して、参照サンプルの全てを使用して平均を計算することができる。或いは参照サンプルの一部だけ（例えば現ブロックの左上の位置に対応する参照サンプル）を使用して平均を計算することができる。

10

【0123】

[0140] 図5に示すように、元の領域内でインタ予測を行いながら、リシェーブされた領域内でイントラ予測（例えばイントラ予測514又は528）を行うことができる。従って、インタ予測では予測ブロックに対して順方向マッピングを適用することができ、順方向マッピング後のルマ予測ブロックを使用して平均を計算する。レイテンシを減らすために、平均は順方向マッピング前の予測ブロックを使用して計算することができる。例えば順方向マッピング前のブロック、順方向マッピング前のブロックのN×N部分、又は順方向マッピング前のブロックの左上サンプルを使用することができる。

20

【0124】

[0141] 本開示の実施形態は、デュアルツリー分割のためのクロマスケールリングを伴う映像コンテンツを処理するための方法を更に提供する。

【0125】

[0142] ルマに対する依存性はハードウェア設計の複雑化を引き起こし得るので、デュアルツリー分割を可能にするイントラコード化されるタイルグループについてクロマスケールリングをオフにすることができる。但し、この制限はコード化効率の損失を引き起こし得る。対応するルマブロックのサンプル値を平均化して $avgY'$ を計算し、区分インデックス Y_{idx} を決定し、クロマスケール因子 $inverse_chroma_scaling[Y_{idx}]$ を得る代わりに、デュアルツリー分割の場合にルマに対する依存性を回避するためにクロマスケール因子をビットストリーム内で明示的にシグナリングすることができる。

30

【0126】

[0143] クロマスケールインデックスは様々なレベルでシグナリングすることができる。例えば図11の表5に示すように、クロマスケールインデックスは、クロマ予測モードと共にコード化単位(CU)レベルでシグナリングすることができる。現クロマブロックのクロマスケール因子を決定するために、構文要素「 $lmcs_scaling_factor_idx$ 」を使用することができる。「 $lmcs_scaling_factor_idx$ 」がない場合、現クロマブロックのクロマスケール因子は浮動小数点数精度で1.0に、又は等価に固定小数点精度で(1 - $CSC_ALE_FP_PREC$)に等しいと推論することができる。「 $lmcs_chroma_scaling_idx$ 」の許容値の範囲はタイルグループレベルで決定され、後で解説する。

40

【0127】

[0144] 「 $lmcs_chroma_scaling_idx$ 」の可能な値にもよるが、とりわけ小さいブロックにとってシグナリングのコストが高くなり得る。従って本開示の一部の実施形態では、図11の表5のシグナリング条件がブロックサイズの条件を追加で含み得る。例えば（イタリック体及び灰色の陰影で強調されている）この構文要素「 $lmcs_chroma_scaling_idx$ 」は、現ブロックが所与の数のクロマサンプルよりも多く含む場合、又は現ブロックが所与の幅Wよりも大きい幅若しくは所与の高さHよりも大きい高さを有する場合にのみ、シグナリングされ得る。より小さいブロックでは、「 $lmcs_chroma_scaling_idx$ 」がシ

50

グナリングされない場合、復号器側でそのクロマスケール因子を決定することができる。一部の実施形態では、クロマスケール因子は浮動小数点数精度で 1 . 0 に設定することができる。一部の実施形態では、タイルグループヘッダレベルでデフォルトの「Imcs_chroma_scaling_idx」値を追加することができる（図 6 の 1 参照）。シグナリングされる「Imcs_chroma_scaling_idx」を有さない小さいブロックは、対応するクロマスケール因子を導出するためにこのタイルグループレベルのデフォルトインデックスを使用することができる。一部の実施形態では、小さいブロックのクロマスケール因子は、スケール因子を明示的にシグナリングしたその近傍（例えば上部又は左側の近傍）から継承することができる。

【 0 1 2 8 】

10

[0145] 構文要素「Imcs_chroma_scaling_idx」を C U レベルでシグナリングすることに加えて、この構文要素は C T U レベルでもシグナリングすることができる。しかし、V V C での最大 C T U サイズが 1 2 8 × 1 2 8 であることを所与とし、C T U レベルで同じスケリングを行うことは粗過ぎる場合がある。従って本開示の一部の実施形態では、固定された粒度を使用してこの構文要素「Imcs_chroma_scaling_idx」をシグナリングすることができる。例えば C T U 内の 1 6 × 1 6 の領域ごとに 1 つの「Imcs_chroma_scaling_idx」がシグナリングされ、その 1 6 × 1 6 の領域内の全てのサンプルに適用される。

【 0 1 2 9 】

20

[0146] 現タイルグループの「Imcs_chroma_scaling_idx」の範囲は、現タイルグループ内で許可されるクロマスケール因子の値の数に依存する。現タイルグループ内で許可されるクロマスケール因子の値の数は、上記で論じたように 6 4 エントリクロマ L U T に基づいて決定され得る。或いは現タイルグループ内で許可されるクロマスケール因子の値の数は、上記で論じたクロマスケール因子の計算を使用して決定することができる。

【 0 1 3 0 】

[0147] 例えば「量子化」方法では、LOG2_n の値を 2 に設定する（即ち「n」を 4 に設定する）ことができ、現タイルグループの区分線形モデル内の各区分のコード語割り当てを次のように設定することができる：{0, 65, 66, 64, 67, 62, 62, 64, 64, 64, 67, 64, 64, 62, 61, 0}。すると 6 4 ~ 6 7 までの任意のコード語の値は同じスケール因子の値（小数精度で 1 . 0 ）を有することができ、6 0 ~ 6 3 までの任意のコード語の値は同じスケール因子の値（小数精度で $60 / 64 = 0.9375$ ）を有することができるので、全タイルグループについて可能なスケール因子の値は 2 つしかない。コード語が一切割り当てられていない 2 つの末端区分ではクロマスケール因子をデフォルトで 1 . 0 に設定する。従ってこの例では、現タイルグループ内のブロックに関して「Imcs_chroma_scaling_idx」をシグナリングするのに 1 ビットで十分である。

30

【 0 1 3 1 】

[0148] 区分線形モデルを使用して可能なクロマスケール因子の値を決定する以外に、符号器はタイルグループヘッダにおいて 1 組のクロマスケール因子の値をシグナリングすることができる。次いでブロックレベルにおいて、その 1 組のクロマスケール因子の値及びブロックの「Imcs_chroma_scaling_idx」の値を使用してブロックのクロマスケール因子の値を決定することができる。

40

【 0 1 3 2 】

[0149] 「Imcs_chroma_scaling_idx」をコード化するために C A B A C コード化を適用することができる。ブロックの C A B A C コンテキストは、ブロックの隣接ブロックの「Imcs_chroma_scaling_idx」に依存し得る。例えば C A B A C コンテキストを形成するために、左側のブロック又は上部のブロックを使用することができる。この「Imcs_chroma_scaling_idx」の構文要素の二値化に関して、「Imcs_chroma_scaling_idx」を二値化するために、V V C のドラフト 4 内の ref_idx_10 及び ref_idx_11 構文要素に適用される同じトランケーティドライス二値化を使用することができる。

【 0 1 3 3 】

50

[0150] 「chroma_scaling_idx」をシグナリングする利点は、符号器がレート歪みコストに関して最良の「lmcs_chroma_scaling_idx」を選択できることである。レート歪み最適化を使用して「lmcs_chroma_scaling_idx」を選択することはコード化効率を改善することができ、そのことはシグナリングコストの増加を相殺することを助け得る。

【0134】

[0151] 本開示の実施形態は、LMCS区分線形モデルのシグナリングを伴う映像コンテンツを処理するための方法を更に提供する。

【0135】

[0152] LMCSの方法は16個の区分を有する区分線形モデルを使用するが、タイルグループ内の「SignaledCW[i]」の固有値の数は16よりもはるかに少ない傾向がある。例えば16個の区分の一部はコード語のデフォルトの数「OrgCW」を使用することができ、16個の区分の一部は互いと同じコード語の数を有することができる。従ってLMCS区分線形モデルをシグナリングする代替的方法是、固有のコード語の数「listUniqueCW[]」をシグナリングし、現区分の「listUniqueCW[]」の要素を示すために区分のそれぞれについてインデックスを送信することを含み得る。

【0136】

[0153] 図12の中で修正済みの構文表を示す。図12の表6では新たな又は修正された構文をイタリック体及び灰色の陰影で強調している。

【0137】

[0154] 開示するシグナリング方法の意味規則は以下の通りであり、変更箇所に下線を引いてある：

resaper_model_min_bin_idxは、リシェーパ構築プロセス内で使用される最小ビン（又は区分）インデックスを規定する。reshape_model_min_bin_idxの値は0～MaxBinIdxまでの範囲にあるものとする。MaxBinIdxの値は15に等しいものとする。

resaper_model_delta_max_bin_idxは、最大許容ビン（又は区分）インデックスMaxBinIdx引く、リシェーパ構築プロセス内で使用される最大ビンインデックスを規定する。reshape_model_max_bin_idxの値はMaxBinIdx-reshape_model_delta_max_bin_idxに等しく設定される。

resaper_model_bin_delta_abs_cw_prec_minus1_plus1は、構文reshape_model_bin_delta_abs_CW[i]の表現に使用されるビット数を規定する。

resaper_model_bin_num_unique_cw_minus1_plus1は、コード語アレイlistUniqueCWのサイズを規定する。

resaper_model_bin_delta_abs_CW[i]は、i番目のビンの絶対デルタコード語値を規定する。

resaper_model_bin_delta_sign_CW_flag[i]は、reshape_model_bin_delta_abs_CW[i]の符号を以下のように規定する：

- reshape_model_bin_delta_sign_CW_flag[i]が0に等しい場合、対応する変数RspDeltaCW[i]は正值である。

- さもなければ（reshape_model_bin_delta_sign_CW_flag[i]が0に等しくない）対応する変数RspDeltaCW[i]は負値である。

reshape_model_bin_delta_sign_CW_flag[i]がない場合、対応する変数RspDeltaCW[i]は0に等しいと推論する。

変数RspDeltaCW[i]は、 $RspDeltaCW[i] = (1 - 2 * reshape_model_bin_delta_sign_CW[i]) * reshape_model_bin_delta_abs_CW[i]$ として導出される。

変数listUniqueCW[0]はOrgCWに等しく設定される。i=1... resaper_model_bin_num_unique_cw_minus1までの変数listUniqueCW[i]は以下のように導出される：

- 変数OrgCWを $(1 - BitDepth) / (MaxBinIdx + 1)$ に等しく設定する。

- listUniqueCW[i] = OrgCW + RspDeltaCW[i-1]

resaper_model_bin_cw_idx[i]は、RspCW[i]を導出するために使用されるアレイlistUniqueCW[]のインデックスを規定する。resaper_model_bin_cw_idx[i]の値は0～

10

20

30

40

50

(reshaper_model_bin_num_unique_cw_minus1+1) までの範囲内にあるものとする。

RspCW[i]は以下のように導出される：

- reshaper_model_min_bin_idx = i = reshaper_model_max_bin_idxが成立する場合、

RspCW[i]=listUniqueCW[reshaper_model_bin_cw_idx[i]].

- さもなければRspCW[i]=0。

BitDepth_yの値が10に等しい場合、RspCW[i]の値は $3^2 \sim 2 \cdot \text{OrgCW} - 1$ の範囲内にあり得る。

【0138】

【0155】 本開示の実施形態は、ブロックレベルにおける条件付きクロマスケールリングを伴う映像コンテンツを処理するための方法を更に提供する。

【0139】

【0156】 図6の表1に示すように、クロマスケールリングが適用されるかどうかはタイルグループレベルでシグナリングされる「tile_group_reshaper_chroma_residual_scale_flag」によって判定することができる。

【0140】

【0157】 しかし、ブロックレベルでクロマスケールリングを適用するかどうかを判定することが有益であり得る。例えば開示する一部の実施形態では、現ブロックにクロマスケールリングが適用されるかどうかを示すために、CUレベルフラグをシグナリングすることができる。CUレベルフラグの存在は、タイルグループレベルフラグ「tile_group_reshaper_chroma_residual_scale_flag」に基づいて条件付けることができる。つまりCUレベルフラグは、クロマスケールリングがタイルグループレベルで許可される場合にのみシグナリングされ得る。符号器は、現ブロックにとってクロマスケールリングが有益かどうかに基づいてクロマスケールリングを使用するかどうかを選ぶことを許可されるが、それは著しいシグナリングオーバーヘッドも招き得る。

【0141】

【0158】 開示する実施形態と合致して、上記のシグナリングオーバーヘッドを回避するために、ブロックにクロマスケールリングが適用されるかどうかはブロックの予測モードに基づいて条件付けることができる。例えばブロックがインタ予測される場合、とりわけその参照ピクチャが時間的距離の点で近い場合は、予測信号がよい傾向にある。従って、残差が非常に小さいと预期されるので、クロマスケールリングをバイパスすることができる。例えば、より高い時間的レベル内のピクチャは時間的距離の点で近い参照ピクチャを有する傾向がある。ブロックに関して、近くの参照ピクチャを使用するピクチャ内でクロマスケールリングを無効化することができる。この条件が満たされるかどうかを判定するために、現ピクチャとブロックの参照ピクチャとの間のピクチャ順序カウンタ（POC）の差を使用することができる。

【0142】

【0159】 一部の実施形態では、全てのインタコード化されるブロックについてクロマスケールリングを無効化することができる。一部の実施形態では、VV規格内で定められる複合イントラ/インタ予測（CIIIP）モードについてクロマスケールリングを無効化することができる。

【0143】

【0160】 VV規格では、CU構文構造「coding_unit()」が、現CU内に何らかの非ゼロ残差係数があるかどうかを示すための構文要素「cu_cbf」を含む。TUレベルでは、TU構文構造「transform_unit()」が、現TU内に何らかの非ゼロクロマ（Cb又はCr）残差係数があるかどうかを示すための構文要素「tu_cbf_cb」及び「tu_cbf_cr」を含む。これらのフラグに基づいてクロマスケールリングプロセスを条件付けることができる。上記で説明したように、非ゼロ残差係数がない場合は、対応するルマクロマスケールリングプロセスの平均化を呼び出すことができる。平均化を呼び出すことにより、クロマスケール

10

20

30

40

50

リングプロセスをバイパスすることができる。

【0144】

[0161] 図13は、映像コンテンツを処理するためのコンピュータによって実装される方法1300の流れ図を示す。一部の実施形態では、方法1300は、コーデック（例えば図2A～図2Bの符号器又は図3A～図3Bの復号器）によって実行され得る。例えばコーデックは、映像シーケンスを符号化し又は別のコードに変換するための機器（例えば機器400）の1つ又は複数のソフトウェア又はハードウェア構成要素として実装することができる。一部の実施形態では、映像シーケンスは、非圧縮映像シーケンス（例えば映像シーケンス202）又は復号される圧縮映像シーケンス（例えば映像ストリーム304）であり得る。一部の実施形態では、映像シーケンスは、機器のプロセッサ（例えばプロセッサ402）に関連する監視装置（例えば図4の映像入力装置）によって捕捉され得る監視映像シーケンスであり得る。映像シーケンスは複数のピクチャを含み得る。機器はピクチャのレベルで方法1300を実行することができる。例えば機器は方法1300内でピクチャを1つずつ処理することができる。別の例では、機器が方法1300内で1度に複数のピクチャを処理することができる。方法1300は以下のようなステップを含み得る。

10

【0145】

[0162] ステップ1302で、ピクチャに関連するクロマブロック及びルマブロックを受信することができる。ピクチャはクロマ成分及びルマ成分に関連し得ることが理解されよう。従って、ピクチャは、クロマサンプルを含むクロマブロック及びルマサンプルを含むルマブロックに関連することができる。

20

【0146】

[0163] ステップ1304で、ルマブロックに関連するルマスケール情報を決定することができる。一部の実施形態では、ルマスケール情報は、ピクチャのデータストリーム内でシグナリングされる構文要素、又はピクチャのデータストリーム内でシグナリングされる構文要素に基づいて導出される変数であり得る。例えばルマスケール情報は、上記の等式の中で記載した「 $\text{reshape_model_bin_delta_sign_CW}[i]$ 及び $\text{reshape_model_bin_delta_abs_CW}[i]$ 」、及び/又は上記の等式の中で記載した「 $\text{SignaledCW}[i]$ 」等を含むことができる。一部の実施形態では、ルマスケール情報は、ルマブロックに基づいて決定される変数を含み得る。例えば平均ルマ値は、ルマブロックに隣接するルマサンプル（ルマブロックの上部のロー内のルマサンプル及びルマブロックの左側のカラム内のルマサンプル等）の平均値を計算することによって決定され得る。

30

【0147】

[0164] ステップ1306で、ルマスケール情報に基づいてクロマスケール因子を決定することができる。

【0148】

[0165] 一部の実施形態では、ルマスケール情報に基づいてルマブロックのルマスケール因子を決定することができる。例えば「 $\text{inverse_chroma_scaling}[i]=((1 - (\text{luma_bit_depth} - \log_2(\text{TOTAL_NUMBER_PIECES}) + \text{CSCALE_FP_PREC})) + (\text{tempCW} - 1)) / \text{tempCW}$ 」の上記の等式によれば、ルマスケール情報（例えば「tempCW」）に基づいてルマスケール因子を決定することができる。次いで、ルマスケール因子の値に基づいてクロマスケール因子を更に決定することができる。例えば、クロマスケール因子はルマスケール因子の値に等しく設定することができる。クロマスケール因子として設定する前に、ルマスケール因子の値に対して更なる計算を適用できることが理解されよう。別の例として、クロマスケール因子は「 $\text{SignaledCW}[\text{Y}_{\text{Idx}}] / \text{OrgCW}$ 」に等しく設定することができる。ルマブロックに関連する平均ルマ値に基づいて現クロマブロックの区分インデックス「 Y_{Idx} 」を決定することができる。

40

【0149】

[0166] ステップ1308で、クロマスケール因子を使用してクロマブロックを処理することができる。例えばクロマブロックのスケールされた残差を生成するために、ク

50

ロマスケール因子を使用してクロマブロックの残差を処理することができる。クロマブロックはC bクロマ成分又はC rクロマ成分であり得る。

【0150】

[0167] 一部の実施形態では、条件が満たされる場合にクロマブロックを処理することができる。例えば条件は、ピクチャに関連する標的コード化単位が非ゼロ残差を有さないこと、又はピクチャに関連する標的変換単位が非ゼロクロマ残差を有さないことを含む得る。標的コード化単位が非ゼロ残差を有さないことは、標的コード化単位の第1のコード化ブロックフラグの値に基づいて判定することができる。標的変換単位が非ゼロクロマ残差を有さないことは、標的変換単位の第1の成分の第2のコード化ブロックフラグの値及び第2の成分の第3のコード化ブロックフラグの値に基づいて判定することができる。例えば第1の成分はC b成分とすることができ、第2の成分はC r成分とすることができ。

10

【0151】

[0168] 方法1300の各ステップは独立した方法として実行できることが理解されよう。例えば、ステップ1308に記載したロマスケール因子を決定するための方法を独立した方法として実行することができる。

【0152】

[0169] 図14は、映像コンテンツを処理するためのコンピュータによって実装される方法1400の流れ図を示す。一部の実施形態では、方法1300は、コーデック（例えば図2A～図2Bの符号器又は図3A～図3Bの復号器）によって実行され得る。例えばコーデックは、映像シーケンスを符号化し又は別のコードに変換するための機器（例えば機器400）の1つ又は複数のソフトウェア又はハードウェア構成要素として実装することができる。一部の実施形態では、映像シーケンスは、非圧縮映像シーケンス（例えば映像シーケンス202）又は復号される圧縮映像シーケンス（例えば映像ストリーム304）であり得る。一部の実施形態では、映像シーケンスは、機器のプロセッサ（例えばプロセッサ402）に関連する監視装置（例えば図4の映像入力装置）によって捕捉され得る監視映像シーケンスであり得る。映像シーケンスは複数のピクチャを含む得る。機器はピクチャのレベルで方法1400を実行することができる。例えば機器は方法1400内でピクチャを1つずつ処理することができる。別の例では、機器が方法1400内で1度に複数のピクチャを処理することができる。方法1400は以下のようなステップを含む得る。

20

30

【0153】

[0170] ステップ1402で、ピクチャに関連するクロマブロック及びルマブロックを受信することができる。ピクチャはクロマ成分及びルマ成分に関連し得ることが理解されよう。従って、ピクチャは、クロマサンプルを含むクロマブロック及びルマサンプルを含むルマブロックに関連することができる。一部の実施形態では、ルマブロックがN×Mのルマサンプルを含む得る。Nはルマブロックの幅とすることができ、Mはルマブロックの高さとすることができ。上記で論じたように、ルマブロックのルマサンプルは標的クロマブロックの区分インデックスを決定するために使用することができる。従って映像シーケンスのピクチャに関連するルマブロックが受信され得る。NとMとが同じ値を有し得ることが理解されよう。

40

【0154】

[0171] ステップ1404で、N及びMの少なくとも1つが閾値を上回ることに応じてN×Mルマサンプルのサブセットを選択することができる。区分インデックスの決定を加速するために、一定の条件が満たされる場合、ルマブロックを「ダウンサンプリング」することができる。換言すれば、ルマブロック内のルマサンプルのサブセットを使用して区分インデックスを決定することができる。一部の実施形態では、一定の条件とはN及びMの少なくとも一方が閾値を上回ることである。一部の実施形態では、閾値はN及びMの少なくとも一方に基づき得る。閾値は2のべき乗とすることができ。例えば閾値は4、8、16等とすることができ。4を例に取り、N又はMが4を上回る場合、ルマサンプルのサブセットを選択することができる。図10の例ではルマブロックの幅及び高さの両方

50

が4の閾値を上回り、従って4×4サンプルのサブセットが選択される。2×8、1×16等のサブセットも処理のために選択できることが理解されよう。

【0155】

[0172] ステップ1406で、N×Mルマサンプルのサブセットの平均値を決定することができる。

【0156】

[0173] 一部の実施形態では、平均値を決定することが、第2の条件が満たされているかどうかを判定すること、及び第2の条件が満たされているという判定に応じてN×Mルマサンプルのサブセットの平均値を決定することを更に含み得る。例えば第2の条件は、ピクチャに関連する標的コード化単位が非ゼロ残差係数を有さないこと又は標的コード化単位が非ゼロクロマ残差係数を有さないことを含み得る。

10

【0157】

[0174] ステップ1408で、平均値に基づくクロマスケール因子を決定することができる。一部の実施形態では、クロマスケール因子を決定するために、クロマブロックの区分インデックスを平均値に基づいて決定することができ、クロマブロックの区分インデックスが第1の条件を満たすかどうかを判定することができ、クロマブロックの区分インデックスが第1の条件を満たすことに応じてクロマスケール因子をデフォルト値に設定することができる。デフォルト値はクロマスケールが適用されないことを示し得る。例えばデフォルト値は小数精度を有する1.0であり得る。デフォルト値に固定小数点近似を適用できることが理解されよう。クロマブロックの区分インデックスが第1の条件を満たさないことに応じて、クロマスケール因子を平均値に基づいて決定することができる。より具体的には、クロマスケール因子を $\text{SignaledCW}[Y_{\text{Idx}}]/\text{OrgCW}$ に設定することができ、標的クロマブロックの区分インデックス「 Y_{Idx} 」は対応するルマブロックの平均値に基づいて決定することができる。

20

【0158】

[0175] 一部の実施形態では、第1の条件はクロマブロックの区分インデックスが、シグナリングされるコード語の最大インデックスよりも大きいこと又はシグナリングされるコード語の最小インデックスよりも小さいことを含み得る。シグナリングされるコード語の最大インデックス及び最小インデックスは以下のように決定することができる。

【0159】

30

[0176] コード語は入力信号（例えばルマサンプル）に基づいて区分線形モデル（例えばLMCS）を使用して生成することができる。上記で論じたように、入力信号のダイナミックレンジは幾つかの区分（例えば16個の区分）に分割することができ、入力信号の各区分を使用してコード語のピンを出力として生成することができる。従って、コード語の各ピンは入力信号の区分に対応するピンインデックスを有し得る。この例では、ピンインデックスの範囲を0～15とすることができる。一部の実施形態では、出力（即ちコード語）の値が最小値（例えば0）と最大値（例えば255）との間であり、最小値と最大値との間の値を有する複数のコード語がシグナリングされ得る。シグナリングされる複数のコード語のピンインデックスを決定することができる。シグナリングされる複数のコード語のピンインデックスのうち、シグナリングされる複数のコード語のピンの最大ピンインデックス及び最小ピンインデックスを更に決定することができる。

40

【0160】

[0177] クロマスケール因子に加えて、方法1400は、シグナリングされる複数のコード語のピンに基づいてルマスケール因子を更に決定することができる。ルマスケール因子は、逆クロマスケール因子として使用することができる。ルマスケール因子を決定するための等式は上記で記載しており、その説明は本明細書では省略する。一部の実施形態では、シグナリングされる複数個の隣接コード語がルマスケール因子を共有する。例えば2個又は4個のシグナリングされる隣接コード語が同じルマスケール因子を共有することができ、そのことはルマスケール因子を決定する負荷を減らし得る。

【0161】

50

【0178】 ステップ1410で、クロマスケール因子を使用してクロマブロックを処理することができる。図5に関して上記で論じたように、複数のクロマスケール因子がタイルグループレベルにおけるクロマスケール因子LUTを構築することができ、標的ブロックの再構築済みクロマ残差に対して復号器側で適用することができる。同様に、クロマスケール因子は符号器側でも適用することができる。

【0162】

【0179】 方法1400の各ステップは独立した方法として実行できることが理解されよう。例えば、ステップ1308に記載したクロマスケール因子を決定するための方法を独立した方法として実行することができる。

【0163】

【0180】 図15は、映像コンテンツを処理するためのコンピュータによって実装される方法1500の流れ図を示す。一部の実施形態では、方法1500は、コーデック（例えば図2A～図2Bの符号器又は図3A～図3Bの復号器）によって実行され得る。例えばコーデックは、映像シーケンスを符号化し又は別のコードに変換するための機器（例えば機器400）の1つ又は複数のソフトウェア又はハードウェア構成要素として実装することができる。一部の実施形態では、映像シーケンスは、非圧縮映像シーケンス（例えば映像シーケンス202）又は復号される圧縮映像シーケンス（例えば映像ストリーム304）であり得る。一部の実施形態では、映像シーケンスは、機器のプロセッサ（例えばプロセッサ402）に関連する監視装置（例えば図4の映像入力装置）によって捕捉され得る監視映像シーケンスであり得る。映像シーケンスは複数のピクチャを含み得る。機器はピクチャのレベルで方法1500を実行することができる。例えば機器は方法1500内でピクチャを1つずつ処理することができる。別の例では、機器が方法1500内で1度に複数のピクチャを処理することができる。方法1500は以下のようなステップを含み得る。

【0164】

【0181】 ステップ1502で、受信映像データ内にクロマスケールインデックスがあるかどうかを判定することができる。

【0165】

【0182】 ステップ1504で、受信映像データ内にクロマスケールインデックスがないと判定することに応じて、受信映像データにクロマスケールリングが適用されないと判定することができる。

【0166】

【0183】 ステップ1506で、受信映像データ内にクロマスケールリングがあると判定することに応じて、クロマスケールインデックスに基づいてクロマスケール因子を決定することができる。

【0167】

【0184】 図16は、映像コンテンツを処理するためのコンピュータによって実装される方法1600の流れ図を示す。一部の実施形態では、方法1600は、コーデック（例えば図2A～図2Bの符号器又は図3A～図3Bの復号器）によって実行され得る。例えばコーデックは、映像シーケンスを符号化し又は別のコードに変換するための機器（例えば機器400）の1つ又は複数のソフトウェア又はハードウェア構成要素として実装することができる。一部の実施形態では、映像シーケンスは、非圧縮映像シーケンス（例えば映像シーケンス202）又は復号される圧縮映像シーケンス（例えば映像ストリーム304）であり得る。一部の実施形態では、映像シーケンスは、機器のプロセッサ（例えばプロセッサ402）に関連する監視装置（例えば図4の映像入力装置）によって捕捉され得る監視映像シーケンスであり得る。映像シーケンスは複数のピクチャを含み得る。機器はピクチャのレベルで方法1600を実行することができる。例えば機器は方法1600内でピクチャを1つずつ処理することができる。別の例では、機器が方法1600内で1度に複数のピクチャを処理することができる。方法1600は以下のようなステップを含み得る。

10

20

30

40

50

【 0 1 6 8 】

[0185] ステップ 1 6 0 2 で、入力映像信号のダイナミックレンジに使用される複数の固有のコード語を受信することができる。

【 0 1 6 9 】

[0186] ステップ 1 6 0 4 で、インデックスを受信することができる。

【 0 1 7 0 】

[0187] ステップ 1 6 0 6 で、複数の固有のコード語の少なくとも 1 つをインデックスに基づいて選択することができる。

【 0 1 7 1 】

[0188] ステップ 1 6 0 8 で、選択済みの少なくとも 1 つのコード語に基づいてクロマスケール因子を決定することができる。

10

【 0 1 7 2 】

[0189] 一部の実施形態では命令を含む非一時的コンピュータ可読記憶媒体も提供され、命令は上記の方法を実行するために装置（開示した符号器及び復号器等）によって実行され得る。一般的な非一時的媒体は、例えばフロッピーディスク、フレキシブルディスク、ハードディスク、ソリッドステートドライブ、磁気テープ若しくは他の任意の磁気データ記憶媒体、CD-ROM、他の任意の光学データ記憶媒体、孔のパターンを有する任意の物理媒体、RAM、PROM、EPROM、フラッシュEPROM若しくは他の任意のフラッシュメモリ、NVRAM、キャッシュ、レジスタ、他の任意のメモリチップ若しくはカートリッジ、及びそれらのもののネットワーク化されたバージョンを含む。装置は、1 つ又は複数のプロセッサ（CPU）、入力／出力インタフェース、ネットワークインタフェース、及び／又はメモリを含み得る。

20

【 0 1 7 3 】

[0190] 上記の実施形態はハードウェア、ソフトウェア（プログラムコード）、又はハードウェアとソフトウェアとの組み合わせによって実装できることが理解されよう。ソフトウェアによって実装される場合、そのソフトウェアは上記のコンピュータ可読媒体の中に記憶することができる。プロセッサによって実行されるとき、ソフトウェアは開示した方法を実行することができる。本開示に記載した計算ユニット及び他の機能ユニットは、ハードウェア、又はソフトウェア、又はハードウェアとソフトウェアとの組み合わせによって実装することができる。上記のモジュール／ユニットの複数のものを 1 つのモジュール／ユニットに組み合わせることができ、上記のモジュール／ユニットのそれぞれを複数のサブモジュール／サブユニットへと更に分けられることも当業者なら理解されよう。

30

【 0 1 7 4 】

[0191] 実施形態は以下の条項を使用して更に記載することができる：

1．映像コンテンツを処理するためのコンピュータ実施方法であって、ピクチャに関連するクロマブロック及びルマブロックを受信すること、ルマブロックに関連するルマスケール情報を決定すること、ルマスケール情報に基づいてクロマスケール因子を決定すること、及びクロマスケール因子を使用してクロマブロックを処理することを含む、方法。

40

2．ルマスケール情報に基づいてクロマスケール因子を決定することが、ルマスケール情報に基づいてルマブロックのルマスケール因子を決定すること、ルマスケール因子の値に基づいてクロマスケール因子を決定することを含む、条項 1 に記載の方法。

3．ルマスケール因子の値に基づいてクロマスケール因子を決定することが、クロマスケール因子をルマスケール因子の値に等しく設定することを含む、条項 2 に記載の方法。

4．クロマスケール因子を使用してクロマブロックを処理することが、第 1 の条件が満たされているかどうかを判定すること、及び

第 1 の条件が満たされているという判定に応じてクロマスケール因子を使用してクロ

50

マブロックを処理すること、又は

第 1 の条件が満たされていないという判定に応じてクロマスケール因子を用いたクロマブロックの処理をバイパスすること

の 1 つを行うこと

を更に含む、条項 1 ~ 3 の何れか 1 つに記載の方法。

5 . 第 1 の条件が、

ピクチャに関連する標的コード化単位が非ゼロ残差を有さないこと、又は

ピクチャに関連する標的変換単位が非ゼロクロマ残差を有さないこと

を含む、条項 4 に記載の方法。

6 . 標的コード化単位が非ゼロ残差を有さないことが、標的コード化単位の第 1 のコード化ブロックフラグの値に基づいて判定され、

10

標的変換単位が非ゼロクロマ残差を有さないことが、標的変換単位の第 1 のクロマ成分の第 2 のコード化ブロックフラグの値及び第 2 のルマクロマ成分の第 3 のコード化ブロックフラグの値に基づいて判定される、

条項 5 に記載の方法。

7 . 第 1 のコード化ブロックフラグの値が 0 であり、

第 2 のコード化ブロックフラグの値及び第 3 のコード化ブロックフラグの値が 0 である、条項 6 に記載の方法。

8 . クロマスケール因子を使用してクロマブロックを処理することが、

クロマスケール因子を使用してクロマブロックの残差を処理すること

20

を含む、条項 1 ~ 7 の何れか 1 つに記載の方法。

9 . 映像コンテンツを処理するための機器であって、

1 組の命令を記憶するメモリと、

メモリに結合され、

ピクチャに関連するクロマブロック及びルマブロックを受信すること、

ルマブロックに関連するルマスケール情報を決定すること、

ルマスケール情報に基づいてクロマスケール因子を決定すること、及び

クロマスケール因子を使用してクロマブロックを処理すること

を機器に行わせるための 1 組の命令を実行するように構成されるプロセッサと

を含む、機器。

30

1 0 . ルマスケール情報に基づいてクロマスケール因子を決定する際、

ルマスケール情報に基づいてルマブロックのルマスケール因子を決定すること、

ルマスケール因子の値に基づいてクロマスケール因子を決定すること

を機器に更に行わせるための 1 組の命令を実行するようにプロセッサが構成される、条項 9 に記載の機器。

1 1 . ルマスケール因子の値に基づいてクロマスケール因子を決定する際、

クロマスケール因子をルマスケール因子の値に等しく設定すること

を機器に更に行わせるための 1 組の命令を実行するようにプロセッサが構成される、条項 1 0 に記載の機器。

1 2 . クロマスケール因子を使用してクロマブロックを処理する際、

40

第 1 の条件が満たされているかどうかを判定すること、及び

第 2 の条件が満たされているという判定に応じてクロマスケール因子を使用してクロマブロックを処理すること、又は

第 2 の条件が満たされていないという判定に応じてクロマスケール因子を用いたクロマブロックの処理をバイパスすること

の 1 つを行うこと

を機器に更に行わせるための 1 組の命令を実行するようにプロセッサが構成される、条項 9 ~ 1 1 の何れか 1 つに記載の機器。

1 3 . 第 1 の条件が、

ピクチャに関連する標的コード化単位が非ゼロ残差を有さないこと、又は

50

ピクチャに関連する標的変換単位が非ゼロクロマ残差を有さないこと
を含む、条項 12 に記載の機器。

14. 標的コード化単位が非ゼロ残差を有さないことが、標的コード化単位の第 1 のコード化ブロックフラグの値に基づいて判定され、

標的変換単位が非ゼロクロマ残差を有さないことが、標的変換単位の第 1 のクロマ成分の第 2 のコード化ブロックフラグの値及び第 2 のクロマ成分の第 3 のコード化ブロックフラグの値に基づいて判定される、

条項 13 に記載の機器。

15. 第 1 のコード化ブロックフラグの値が 0 であり、

第 2 のコード化ブロックフラグの値及び第 3 のコード化ブロックフラグの値が 0 である、
条項 14 に記載の機器。 10

16. クロマスケール因子を使用してクロマブロックを処理する際、

クロマスケール因子を使用してクロマブロックの残差を処理すること

を機器に更に行わせるための 1 組の命令を実行するようにプロセッサが構成される、条項 9 ~ 15 の何れか 1 つに記載の機器。

17. 映像コンテンツを処理するための方法を装置に行わせるための、装置の 1 つ又は複数のプロセッサによって実行可能な 1 組の命令を記憶する非一時的コンピュータ可読記憶媒体であって、その方法は、

ピクチャに関連するクロマブロック及びルマブロックを受信すること、

ルマブロックに関連するルマスケール情報を決定すること、 20

ルマスケール情報に基づいてクロマスケール因子を決定すること、及び

クロマスケール因子を使用してクロマブロックを処理すること

を含む、非一時的コンピュータ可読記憶媒体。

18. 映像コンテンツを処理するためのコンピュータ実施方法であって、

ピクチャに関連するクロマブロック及びルマブロックを受信することであって、ルマブロックは $N \times M$ のルマサンプルを含む、受信すること、

N 及び M の少なくとも 1 つが閾値を上回ることに応じて $N \times M$ ルマサンプルのサブセットを選択すること、

$N \times M$ ルマサンプルのサブセットの平均値を決定すること、

平均値に基づいてクロマスケール因子を決定すること、及び 30

クロマスケール因子を使用してクロマブロックを処理すること

を含む、方法。

19. 映像コンテンツを処理するためのコンピュータ実施方法であって、

受信映像データ内にクロマスケールインデックスがあるかどうかを判定すること、

受信映像データ内にクロマスケールインデックスがないと判定することに応じて、受信映像データにクロマスケールリングが適用されないと判定すること、及び

受信映像データ内にクロマスケールリングがあると判定することに応じて、クロマスケールインデックスに基づいてクロマスケール因子を決定すること

を含む、方法。

20. 映像コンテンツを処理するためのコンピュータ実施方法であって、 40

入力映像信号のダイナミックレンジに使用される複数の固有のコード語を受信すること、インデックスを受信すること、

複数の固有のコード語の少なくとも 1 つをインデックスに基づいて選択すること、及び

選択済みの少なくとも 1 つのコード語に基づいてクロマスケール因子を決定すること

を含む、方法。

【0175】

[0192] コンピュータ可読プログラムコードを使用することによって上記の方法を実装することに加えて、上記の方法は論理ゲート、スイッチ、ASIC、プログラム可能論理コントローラ、及び埋め込みマイクロコントローラの形で実装することもできる。従って、かかるコントローラはハードウェア構成要素と見なすことができ、コントローラ内に含 50

まれ様々な機能を実装するように構成される機器もハードウェア構成要素内の構造と見なすことができる。又は様々な機能を実装するように構成される機器は、方法を実装するように構成されるソフトウェアモジュール及びハードウェア構成要素内の構造の両方と見なすことさえできる。

【0176】

[0193] 本開示はコンピュータによって実行されるコンピュータ実行可能命令、例えばプログラムモジュールの全般的な脈絡で説明することができる。概してプログラムモジュールは、特定のタスクを実行し又は特定の抽象データ型を実装するために使用されるルーチン、プログラム、オブジェクト、アセンブリ、データ構造、クラス等を含む。本開示の実施形態は分散型計算環境内で実装することもできる。分散型計算環境では、通信ネットワークによって接続される遠隔処理装置を使用することによってタスクが実行される。分散型計算環境では、プログラムモジュールが記憶装置を含む局所及び遠隔コンピュータ記憶媒体内にあり得る。

10

【0177】

[0194] 本明細書の「第1の」及び「第2の」等の関係語は或るエンティティ又は操作を別のエンティティ又は操作と区別するために使用するに過ぎず、それらのエンティティ又は操作間の如何なる実際の関係又は順序も必要とせず又は含意しないことに留意すべきである。更に、「comprising（含む）」、「having（有する）」、「containing（含む）」、及び「including（含む）」並びに他の同様の形式の言葉は意味の点で等価であることを意図し、これらの言葉の何れか1つの後に続くアイテムがかかるアイテムの網羅的列挙であることを意図しておらず、又は列挙するアイテムだけに限定されることを意図していない点で非限定的であることを意図する。

20

【0178】

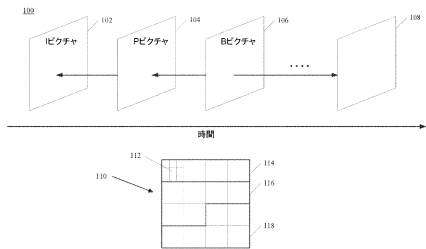
[0195] 上記の明細書では、実装形態ごとに変わり得る多数の具体的詳細に関して実施形態を説明してきた。記載した実施形態の一定の適応及び修正を加えることができる。本明細書を検討し、本明細書で明らかにした開示を实践することで他の実施形態が当業者に明らかになり得る。本明細書及び例は専ら例として検討され、本開示の真の範囲及び趣旨は添付の特許請求の範囲によって示すことを意図する。図中に示すステップの順序は例示目的に過ぎず、或る特定のステップの順序に限定されることは意図しない。そのため、それらのステップは同じ方法を実装しながら異なる順序で実行できることを当業者なら理解することができる。

30

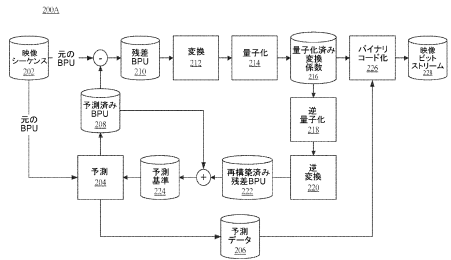
40

50

【図面】
【図 1】

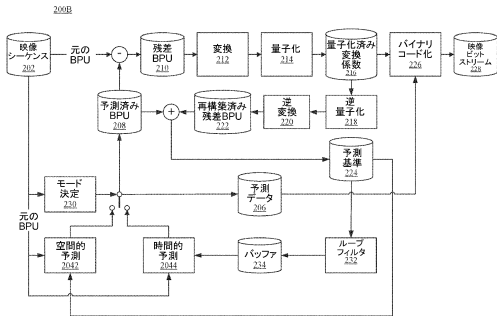


【図 2 A】

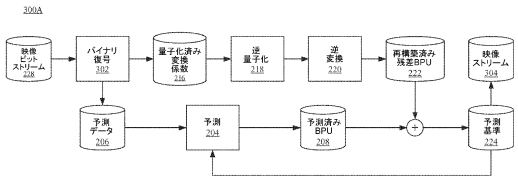


10

【図 2 B】



【図 3 A】



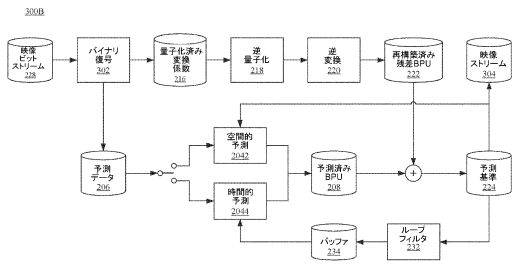
20

30

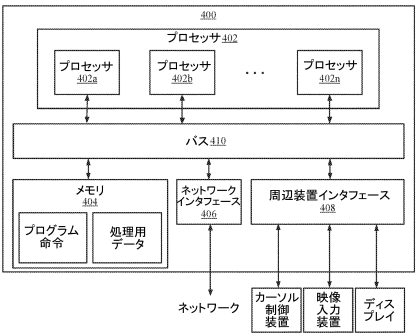
40

50

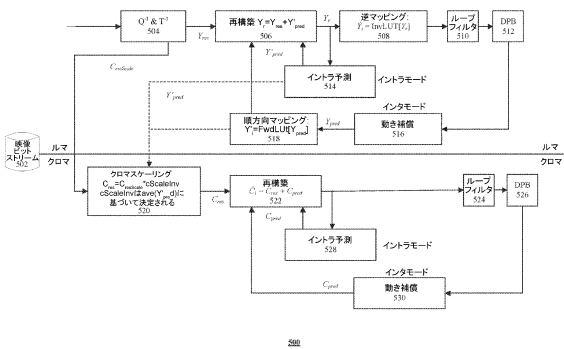
【図 3 B】



【図 4】



【図 5】



【図 6】

表1

tile_group_header()	記述子
...	
if (sps_resampler_enabled_flag) {	
tile_group_resampler_model_present_flag	u(1)
if (tile_group_resampler_model_present_flag) {	
tile_group_resampler_model()	
tile_group_resampler_enable_flag	u(1)
if (tile_group_resampler_enable_flag && !(qbit_dual_tree_intra_flag && tile_group_type == 1)) {	
tile_group_resampler_chroma_residual_scale_flag	u(1)
}	
}	
byte_alignment()	
}	

LMCS区分線形モデルのためのタイルグループレベルの構文表

【図 7】

【図 8】

表2

	記述子
tile_group_resaper_model ()	
resaper_model_min_bin_idx	u(v)
resaper_model_delta_max_bin_idx	u(v)
resaper_model_bin_delta_abs_cw_minval	u(v)
for (i = resaper_model_min_bin_idx; i <= resaper_model_max_bin_idx; i++) {	
resaper_model_bin_delta_abs_cw [i]	u(v)
if (resaper_model_bin_delta_abs_cw [i] > 0)	
resaper_model_bin_delta_sign_cw_flag [i]	u(1)
}	

LMCS区分線形モデルのためのタイルグループレベルの構文表

表3

	記述子
coding_tree_init () {	
xCb = (CbAddrInRs % PicWidthInCbY) << CbLog2SizeY	
yCb = (CbAddrInRs / PicWidthInCbY) << CbLog2SizeY	
if (tile_group_sao_luma_flag tile_group_sao_chroma_flag)	
saoL_xCb >> CbLog2SizeY, yCb >> CbLog2SizeY	
if (tile_group_alf_enabled_flag) {	
alf_cbf_flag [0] [xCb >> Log2CbSize] [yCb >> Log2CbSize]	ae(v)
if (alf_chroma_idc == 1 alf_chroma_idc == 3)	
alf_cbf_flag [1] [xCb >> Log2CbSize] [yCb >> Log2CbSize]	ae(v)
if (alf_chroma_idc == 2 alf_chroma_idc == 3)	
alf_cbf_flag [2] [xCb >> Log2CbSize] [yCb >> Log2CbSize]	ae(v)
}	
if ((tile_group_type == 1 CurrPicIsOnlyRef) && qbt_dual_tree_intrn_flag)	
dual_tree_implicit_qt_split (xCb, yCb, CbLog2SizeY, 0)	
else	
coding_quadtree (xCb, yCb, CbLog2SizeY, 0, SINGLE_TREE)	
}	

コード化ツリー単位の構文構造

【図 9】

【図 10】

表4

	記述子
dual_tree_implicit_qt_split (x0, y0, log2CbSize, cqtDepth) {	
if (log2CbSize > 6) {	
x1 = x0 + (1 << (log2CbSize - 1))	
y1 = y0 + (1 << (log2CbSize - 1))	
dual_tree_implicit_qt_split (x1, y1, log2CbSize - 1, cqtDepth + 1)	
if (x1 < pic_width_in_luma_samples)	
dual_tree_implicit_qt_split (x1, y0, log2CbSize - 1, cqtDepth + 1)	
if (y1 < pic_height_in_luma_samples)	
dual_tree_implicit_qt_split (x0, y1, log2CbSize - 1, cqtDepth + 1)	
if (x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples)	
dual_tree_implicit_qt_split (x1, y1, log2CbSize - 1, cqtDepth + 1)	
} else {	
coding_quadtree (x0, y0, log2CbSize, cqtDepth, DUAL_TREE_LUMA)	
coding_quadtree (x0, y0, log2CbSize, cqtDepth, DUAL_TREE_CHROMA)	
}	

デュアルツリー分割の構文構造

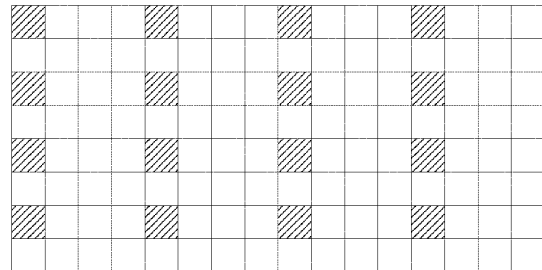


FIG. 10

10

20

30

40

50

【図 1 1】

【図 1 2】

表5

	記述子
coding_unit(x0, y0, cbWidth, cbHeight, treeType) { ... if(CuPredMode[x0][y0] == MODE_INTRA) { ... if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA) { intra_chroma_pred_mode[x0][y0] if(tile_group_reshaper_chroma_residual_scale_flag) lmcv_chroma_scaling_idx[x0][y0] } }	u(v) u(v) u(v) u(v) u(v) u(v)

コード化ツリー単位の構文構造

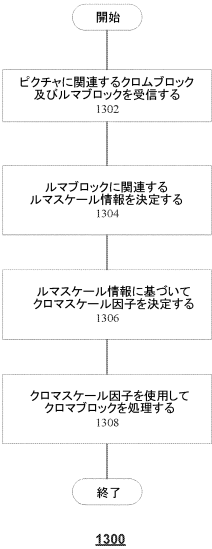
表6

	記述子
tile_group_reshaper_model() { reshaper_model_min_bin_idx reshaper_model_delta_max_bin_idx reshaper_model_bin_delta_abs_cw_prec_minus1 reshaper_model_bin_num_unique_cw_minus1 for(i = 0; i <= reshaper_model_bin_num_unique_cw_minus1; i++) { reshaper_model_bin_delta_abs_CW[i] if(reshaper_model_bin_delta_abs_CW[i] > 0) reshaper_model_bin_delta_sign_CW_flag[i] } for(i = reshaper_model_min_bin_idx; i <= reshaper_model_max_bin_idx; i++) reshaper_model_bin_cw_idx[i] }	u(v) u(v) u(v) u(v) u(v) u(v) u(v) u(v) u(v)

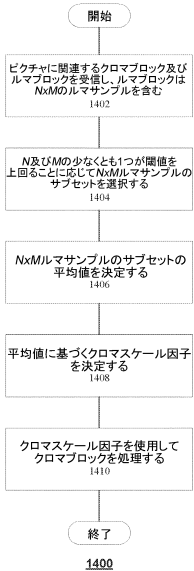
タイルグループレベルにおけるLMCS区分線形モデルの修正されたシグナリング

【図 1 3】

【図 1 4】



1300



1400

【図 1 3】

【図 1 4】

10

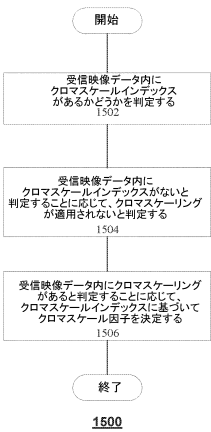
20

30

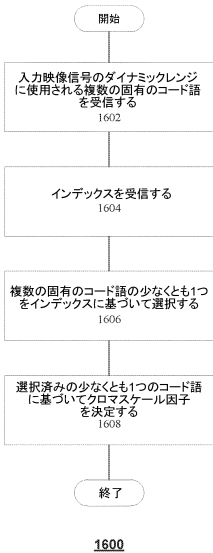
40

50

【 図 1 5 】



【 図 1 6 】



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)

前置審査

(72)発明者 イエ, ヤン

アメリカ合衆国, カリフォルニア州 9 4 4 0 2, サン マテオ, サウス エル カミーノ レアル
4 0 0, スイート 4 0 0, アリババ グループ リーガル デパートメント

(72)発明者 チェン, ジエ

アメリカ合衆国, カリフォルニア州 9 4 4 0 2, サン マテオ, サウス エル カミーノ レアル
4 0 0, スイート 4 0 0, アリババ グループ リーガル デパートメント

(72)発明者 リャオ, ルーリン

アメリカ合衆国, カリフォルニア州 9 4 4 0 2, サン マテオ, サウス エル カミーノ レアル
4 0 0, スイート 4 0 0, アリババ グループ リーガル デパートメント

審査官 田部井 和彦

(56)参考文献 国際公開第 2 0 2 0 / 1 5 6 5 2 9 (WO , A 1)

特表 2 0 2 2 - 5 1 7 6 8 3 (J P , A)

特表 2 0 1 6 - 5 4 1 1 6 4 (J P , A)

国際公開第 2 0 1 8 / 0 1 6 3 8 1 (WO , A 1)

Benjamin Bross et al. , Versatile Video Coding (Draft 4) [online] , JVET-M1001-v5 (JVET-M1001-v5.docx) , [2024年2月26日検索] , インターネット <URL: https://jvet-experts.org/doc_end_user/documents/13_Marrakech/wg11/JVET-M1001-v5.zip> , 2019年02月27日 , pp.49-54,95-98,208Taoran Lu et al. , CE12-2: HDR In-loop Reshaping [online] , JVET-L0245-v2 (JVET-L0245_CE12-2_HDR_inloop_resaping_v2.docx) , [2024年2月26日検索] , インターネット <URL: https://jvet-experts.org/doc_end_user/documents/12_Macao/wg11/JVET-L0245-v2.zip> , 2018年09月29日Jianle Chen et al. , Algorithm description for Versatile Video Coding and Test Model 4 (VTM 4) [online] , JVET-M1002-v1 (JVET-M1002-v1.docx) , [2024年2月26日検索] , インターネット <URL: https://jvet-experts.org/doc_end_user/documents/14_Geneva/wg11/JVET-N1001-v10.zip> , 2019年02月16日 , pp.1-7,54-58Edouard Franois et al. , Suggested luma mapping with chroma scaling modifications in N02 20/N0389/N0477 [online] , JVET-N0806 (JVET-N0806v1.docx) , [2024年2月26日検索] , インターネット <URL: https://jvet-experts.org/doc_end_user/documents/14_Geneva/wg11/JVET-N0806-v3.zip> , 2019年03月23日

(58)調査した分野 (Int.Cl. , D B 名)

H 0 4 N 1 9 / 5 0

H 0 4 N 1 9 / 1 0 2

H 0 4 N 1 9 / 1 3 6

H 0 4 N 1 9 / 1 8 6

H 0 4 N 1 9 / 7 0