(19)日本国特許庁(JP)

# (12)特許公報(B2)

(11)特許番号
特許第7629450号
(P7629450)

(45)発行日　令和7年2月13日(2025.2.13)

(24)登録日　令和7年2月4日(2025.2.4)

(51)国際特許分類

| | | FI | |
|---|---|---|---|
| *H04N* | *19/126 (2014.01)* | H04N | 19/126 |
| *H04N* | *19/157 (2014.01)* | H04N | 19/157 |
| *H04N* | *19/176 (2014.01)* | H04N | 19/176 |
| *H04N* | *19/186 (2014.01)* | H04N | 19/186 |
| *H04N* | *19/70 (2014.01)* | H04N | 19/70 |

請求項の数　18　（全42頁）　最終頁に続く

(21)出願番号　特願2022-514463(P2022-514463)
(86)(22)出願日　令和2年8月21日(2020.8.21)
(65)公表番号　特表2022-548825(P2022-548825A)
(43)公表日　令和4年11月22日(2022.11.22)
(86)国際出願番号　PCT/US2020/047411
(87)国際公開番号　WO2021/055138
(87)国際公開日　令和3年3月25日(2021.3.25)
審査請求日　令和5年8月2日(2023.8.2)
(31)優先権主張番号　62/903,251
(32)優先日　令和1年9月20日(2019.9.20)
(33)優先権主張国・地域又は機関
米国(US)

(73)特許権者　511050697
アリババ グループ ホウルディング リミテッド
英国領ケイマン諸島 グランド ケイマン ジョージ タウン ピーオーボックス 847 ワン キャピタル プレイス フォース フロア
(74)代理人　100079108
弁理士　稲葉 良幸
(74)代理人　100109346
弁理士　大貫 敏史
(74)代理人　100117189
弁理士　江口 昭彦
(74)代理人　100134120
弁理士　内藤 和彦

最終頁に続く

(54)【発明の名称】　映像処理における量子化パラメータのシグナリング

(57)【特許請求の範囲】

【請求項１】

映像ストリームのための１つ又は複数のピクチャを出力するために、ビットストリームを復号する方法であって、

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用いて１つ又は複数のピクチャを復号すること

を含み、前記復号することは、

コード化ブロックの第１のパラメータを決定すること、

デルタ量子化パラメータ（ＱＰ）値又はクロマＱＰオフセット値に関連する１つ又は複数の第２のパラメータを前記第１のパラメータに従って決定すること、及び

前記デルタＱＰ値又は前記クロマＱＰオフセット値の少なくとも１つを前記１つ又は複数の第２のパラメータに従って決定すること

を含み、

前記第１のパラメータの最大値は、

コード化ツリー単位のルマコード化ツリーブロックのサイズ；

イントラ予測スライス又はインター予測スライス内の前記コード化ツリー単位の４分木分割から生じるルマリーフブロック内のルマサンプルの最小サイズ；及び

前記イントラ予測スライス又は前記インター予測スライス内の４分木の葉のマルチタイプツリー分割から生じるコード化単位の最大階層深度；

に基づいて決定される、方法。
【請求項２】
前記コード化ブロックの前記第１のパラメータを決定することは、
前記コード化ブロックがイントラ予測スライス又はインター予測スライスに関連するかどうかを判定すること、及び
前記コード化ブロックが前記イントラ予測スライスに関連することに応じて、前記第１のパラメータを前記イントラ予測スライスに関連するパラメータであるように決定すること、又は
前記コード化ブロックが前記インター予測スライスに関連することに応じて、前記第１のパラメータを前記インター予測スライスに関連するパラメータであるように決定すること
を含む、請求項１に記載の方法。
【請求項３】
前記第１のパラメータが、前記コード化ブロックに関連するスライスヘッダ又はピクチャヘッダの少なくとも１つの中でシグナリングされる、請求項１に記載の方法。
【請求項４】
前記コード化ブロックが前記イントラ予測スライスに関連し、前記方法が、
前記第１のパラメータの前記最大値は、前記イントラ予測スライス内に基づいて決定される、請求項３に記載の方法。
【請求項５】
前記コード化ブロックが前記インター予測スライスに関連し、前記方法が、
前記第１のパラメータの前記最大値は、前記インター予測スライス内に基づいて決定される、請求項３に記載の方法。
【請求項６】
前記デルタＱＰ値に基づいてルマＱＰ値を決定すること、
前記クロマＱＰオフセット値に基づいてクロマＱＰ値を決定すること、及び
前記ルマＱＰ値及び前記クロマＱＰ値に基づいて前記コード化ブロックを処理すること
を更に含む、請求項１に記載の方法。
【請求項７】
映像シーケンスをビットストリームに符号化する方法であって、前記方法は、
映像シーケンスを受信すること、
前記映像シーケンスの１つ又は複数のピクチャを符号化すること、及び
ビットストリームを生成すること
を含み、前記符号化することは、
コード化ブロックの第１のパラメータを決定すること、
デルタ量子化パラメータ（ＱＰ）値又はクロマＱＰオフセット値に関連する１つ又は複数の第２のパラメータを前記第１のパラメータに従って決定すること、及び
前記デルタＱＰ値又は前記クロマＱＰオフセット値の少なくとも１つを前記１つ又は複数の第２のパラメータに従って決定すること
を含み、
前記第１のパラメータの最大値は、
コード化ツリー単位のルマコード化ツリーブロックのサイズ；
イントラ予測スライス又はインター予測スライス内の前記コード化ツリー単位の４分木分割から生じるルマリーフブロック内のルマサンプルの最小サイズ；及び
前記イントラ予測スライス又は前記インター予測スライス内の４分木の葉のマルチタイプツリー分割から生じるコード化単位の最大階層深度；
に基づいて決定される、方法。
【請求項８】
前記コード化ブロックの前記第１のパラメータを決定することは、
前記コード化ブロックがイントラ予測スライス又はインター予測スライスに関連するかどうかを判定すること、及び

前記コード化ブロックが前記イントラ予測スライスに関連することに応じて、前記第１のパラメータを前記イントラ予測スライスに関連するパラメータであるように決定すること、又は
前記コード化ブロックが前記インター予測スライスに関連することに応じて、前記第１のパラメータを前記インター予測スライスに関連するパラメータであるように決定すること
を含む、請求項７に記載の方法。
【請求項９】
前記第１のパラメータが、前記コード化ブロックに関連するスライスヘッダ又はピクチャヘッダの少なくとも１つの中でシグナリングされる、請求項７に記載の方法。
【請求項１０】
前記コード化ブロックが前記イントラ予測スライスに関連し、前記方法が、
前記第１のパラメータの前記最大値は、前記イントラ予測スライス内に基づいて決定される、請求項９に記載の方法。
【請求項１１】
前記コード化ブロックが前記インター予測スライスに関連し、前記方法が、
前記第１のパラメータの前記最大値は、前記インター予測スライス内に基づいて決定される、請求項９に記載の方法。
【請求項１２】
前記デルタＱＰ値に基づいてルマＱＰ値を決定すること、
前記クロマＱＰオフセット値に基づいてクロマＱＰ値を決定すること、及び
前記ルマＱＰ値及び前記クロマＱＰ値に基づいて前記コード化ブロックを処理すること
を更に含む、請求項７に記載の方法。
【請求項１３】
映像シーケンスのビットストリームを記憶するための方法であって、前記方法が、
映像シーケンスを受信すること、
前記映像シーケンスの１つ又は複数のピクチャを符号化すること、
ビットストリームを生成すること、及び
前記ビットストリームを非一時的コンピュータ可読媒体に記憶すること
を含み、前記符号化することは、
コード化ブロックの第１のパラメータを決定すること、
デルタ量子化パラメータ（ＱＰ）値又はクロマＱＰオフセット値に関連する１つ又は複数の第２のパラメータを前記第１のパラメータに従って決定すること、及び
前記デルタＱＰ値又は前記クロマＱＰオフセット値の少なくとも１つを前記１つ又は複数の第２のパラメータに従って決定すること
を含み、
前記第１のパラメータの最大値は、
コード化ツリー単位のルマコード化ツリーブロックのサイズ；
イントラ予測スライス又はインター予測スライス内の前記コード化ツリー単位の４分木分割から生じるルマリーフブロック内のルマサンプルの最小サイズ；及び
前記イントラ予測スライス又は前記インター予測スライス内の４分木の葉のマルチタイプツリー分割から生じるコード化単位の最大階層深度；
に基づいて決定される、方法。
【請求項１４】
前記コード化ブロックの前記第１のパラメータを決定することは、
前記コード化ブロックがイントラ予測スライス又はインター予測スライスに関連するかどうかを判定すること、及び
前記コード化ブロックが前記イントラ予測スライスに関連することに応じて、前記第１のパラメータを前記イントラ予測スライスに関連するパラメータであるように決定すること、又は
前記コード化ブロックが前記インター予測スライスに関連することに応じて、前記第１

のパラメータを前記インター予測スライスに関連するパラメータであるように決定すること
を含む、請求項１３に記載の方法。
【請求項１５】
前記第１のパラメータが、前記コード化ブロックに関連するスライスヘッダ又はピクチャヘッダの少なくとも１つの中でシグナリングされる、請求項１３に記載の方法。
【請求項１６】
前記コード化ブロックが前記イントラ予測スライスに関連し、前記方法が、
前記第１のパラメータの前記最大値は、前記イントラ予測スライス内に基づいて決定される、請求項１５に記載の方法。
【請求項１７】
前記コード化ブロックが前記インター予測スライスに関連し、前記方法が、
前記第１のパラメータの前記最大値は、前記インター予測スライス内に基づいて決定される 請求項１５に記載の方法。
【請求項１８】
前記方法が、
前記デルタＱＰ値に基づいてルマＱＰ値を決定すること、
前記クロマＱＰオフセット値に基づいてクロマＱＰ値を決定すること、及び
前記ルマＱＰ値及び前記クロマＱＰ値に基づいて前記コード化ブロックを処理すること
を更に含む、請求項１３に記載の方法。
【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、参照によりその全体を本明細書に援用する、２０１９年９月２０日に出願された米国仮特許出願第６２／９０３，２５１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、量子化パラメータと共に映像コンテンツを処理するための方法及びシステムに関する。
【背景技術】
【０００３】
背景
［0003］ 映像は、視覚情報を捕捉する静的ピクチャ（又は「フレーム」）の組である。記憶メモリ及び伝送帯域幅を減らすために、映像は、記憶又は伝送前に圧縮し、表示前に解凍することができる。圧縮プロセスは、通常、符号化と呼ばれ、解凍プロセスは、通常、復号と呼ばれる。最も一般的には、予測、変換、量子化、エントロピーコード化及びインループフィルタリングに基づく規格化された映像コード化技術を使用する様々な映像コード化形式がある。特定の映像コード化形式を指定するHigh Efficiency Video Coding（ＨＥＶＣ／Ｈ．２６５）規格、Versatile Video Coding（ＶＶＣ／Ｈ．２６６）規格、ＡＶＳ規格等の映像コード化規格が規格化組織によって策定されている。一層進化した映像コード化技術が映像規格に採用されるにつれて、新たな映像コード化規格のコード化効率が一層高くなる。
【発明の概要】
【課題を解決するための手段】
【０００４】
本開示の概要
［0004］ 本開示の実施形態は、コード化された映像データを含むビットストリームを受信すること、コード化ブロックの第１のパラメータを決定すること、デルタ量子化パラメータ（ＱＰ）値又はクロマＱＰオフセット値に関連する１つ又は複数の第２のパラメータを第１のパラメータに従って決定すること、及びデルタＱＰ値又はクロマＱＰオフセット

値の少なくとも１つを１つ又は複数の第２のパラメータに従って決定することを含む、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施される方法を提供する。

【０００５】

[0005] 本開示の実施形態は、１組の命令を記憶するメモリと、少なくとも１つのプロセッサとを含む映像コンテンツを処理するためのシステムも提供し、少なくとも１つのプロセッサは、コード化された映像データを含むビットストリームを受信すること、コード化ブロックの第１のパラメータを決定すること、デルタ量子化パラメータ（ＱＰ）値又はクロマＱＰオフセット値に関連する１つ又は複数の第２のパラメータを第１のパラメータに従って決定すること、及びデルタＱＰ値又はクロマＱＰオフセット値の少なくとも１つを１つ又は複数の第２のパラメータに従って決定することをシステムに行わせるように、１組の命令を実行するように構成される。

【０００６】

[0006] 本開示の実施形態は、コンピュータシステムの少なくとも１つのプロセッサによって実行可能な命令を記憶する非一時的コンピュータ可読媒体も提供し、命令を実行することは、コード化映像データを含むビットストリームを受信すること、コード化ブロックの第１のパラメータを決定すること、デルタ量子化パラメータ（ＱＰ）値又はクロマＱＰオフセット値に関連する１つ又は複数の第２のパラメータを第１のパラメータに従って決定すること、及びデルタＱＰ値又はクロマＱＰオフセット値の少なくとも１つを１つ又は複数の第２のパラメータに従って決定することを含む方法をコンピュータシステムに行わせる。

【０００７】

図面の簡単な説明

[0007] 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に示す。図中に示す様々な特徴は、縮尺通りに描かれていない。

【図面の簡単な説明】

【０００８】

【図１】[0008]本開示の実施形態と合致する、例示的な映像シーケンスの構造を示す。

【図２Ａ】[0009]本開示の実施形態と合致する、ハイブリッド映像コード化システムの例示的な符号化プロセスの概略図を示す。

【図２Ｂ】[0010]本開示の実施形態と合致する、ハイブリッド映像コード化システムの別の例示的な符号化プロセスの概略図を示す。

【図３Ａ】[0011]本開示の実施形態と合致する、ハイブリッド映像コード化システムの例示的な復号プロセスの概略図を示す。

【図３Ｂ】[0012]本開示の実施形態と合致する、ハイブリッド映像コード化システムの別の例示的な復号プロセスの概略図を示す。

【図４】[0013]本開示の実施形態と合致する、映像を符号化又は復号するための例示的な機器のブロック図である。

【図５】[0014]本開示の実施形態と合致する、コード化単位（ＣＵ）デルタ量子化パラメータ（ＱＰ）のためのピクチャパラメータセット（ＰＰＳ）の一例を示す。

【図６Ａ】[0015]本開示の実施形態と合致する、ＣＵデルタＱＰのためのコード化ツリーレベル構文の一例を示す。

【図６Ｂ】[0015]本開示の実施形態と合致する、ＣＵデルタＱＰのためのコード化ツリーレベル構文の一例を示す。

【図６Ｃ】[0015]本開示の実施形態と合致する、ＣＵデルタＱＰのためのコード化ツリーレベル構文の一例を示す。

【図７Ａ】[0016]本開示の実施形態と合致する、ＣＵデルタＱＰのための変換単位レベル構文の一例を示す。

【図７Ｂ】[0016]本開示の実施形態と合致する、ＣＵデルタＱＰのための変換単位レベル構文の一例を示す。

【図７Ｃ】[0016]本開示の実施形態と合致する、ＣＵデルタＱＰのための変換単位レベル

構文の一例を示す。
【図８】[0017]本開示の実施形態と合致する、スライスヘッダ構文の一例を示す。
【図９】[0018]本開示の実施形態と合致する、スライスヘッダ構文の別の例を示す。
【図１０】[0019]本開示の実施形態と合致する、スライスヘッダ構文の更に別の例を示す。
【図１１】[0020]本開示の実施形態と合致する、ピクチャヘッダ構文の別の例を示す。
【図１２】[0021]本開示の実施形態と合致する、cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivのためのＰＰＳ構文の一例を示す。
【図１３】[0022]本開示の実施形態と合致する、cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivのためのスライスヘッダ構文の一例を示す。
【図１４】[0023]本開示の実施形態と合致する、sps_max_mtt_depth_lumaのための構文の一例を示す。
【図１５】[0024]本開示の実施形態と合致する、pps_max_mtt_depth_lumaのための構文の一例を示す。
【図１６】[0025]本開示の実施形態と合致する、cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivのためのＳＰＳ構文の一例を示す。
【図１７】[0026]本開示の実施形態と合致する、cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivのためのスライスヘッダ構文の一例を示す。
【図１８】[0027]本開示の実施形態と合致する、sps_max_mtt_depth_lumaのための構文の一例を示す。
【図１９】[0028]本開示の実施形態と合致する、sps_max_mtt_depth_lumaのための構文の別の例を示す。
【図２０】[0029]本開示の実施形態と合致する、pps_max_mtt_depth_lumaのための構文の一例を示す。
【図２１】[0030]本開示の実施形態と合致する、映像コンテンツを処理するための例示的なコンピュータによって実装される方法の流れ図である。
【発明を実施するための形態】
【０００９】
[0031] 映像コード化システムは、デジタル映像信号を圧縮するために、例えば消費される記憶空間を減らすか、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ために多くの場合に使用される。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、１９２０×１０８０ピクセルの解像度を有する）高精細度（ＨＤ）映像の人気が高まるにつれて、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継続的に求められている。
【００１０】
[0032] 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択し、ＨＤ映像は、かかる情報を捕捉するために、１フレーム当たりで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、常時監視を行い、それは、映像データを記憶する場合に記憶システムにとって大きい課題となり得る。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。
【００１１】
[0033] 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静止ピクチャ（又は「フレーム」）の組である。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。

【００１２】

［0034］ 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され得る。圧縮のためのモジュールを一般に「符号器」と呼び、解凍のため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、解凍された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。

【００１３】

［0035］ 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。

【００１４】

［0036］ 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重要である。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。

【００１５】

［0037］ 別のピクチャを参照することなくコード化されるピクチャ（即ちそのような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。

【００１６】

［0038］ 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、ほぼ静的であるため、符号化映像の全体的なビットレートを下げる１つの方法は、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。

【００１７】

[0039] しかし、符号化映像内において、Ｉピクチャは、一般に主要なものではないため、Ｉピクチャをより少なく使用する改善策は、些細なものであり得る。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である場合があり、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る。

【００１８】

[0040] 本開示は、適応解像度変更（ＡＲＣ）を使用して映像コンテンツを処理するための方法、機器、及びシステムを提供する。フェーズの丸め処理によって生じる不正確なフェーズと異なり、本開示の実施形態は精度を保ちながらアルゴリズム及びハードウェアの複雑さを減らすために、固定フェーズ補間に基づくピクセルリファインプロセスを提供する。

【００１９】

[0041] 図１は、本開示の実施形態に合致する、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であり得る。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。

【００２０】

[0042] 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、タイムラインに沿って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との間に更に多くのピクチャがある。図１では、ピクチャ１０２は、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、そのピクチャの直前又は直後になくてもよい。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。

【００２１】

[0043] 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せず、それは、かかるタスクが計算的に複雑であるためである。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１０２～１０８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、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、コード化の効率と基本処理単位内で保とうとする詳細度と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。

【００２２】
［0044］基本処理単位は、コンピュータメモリ内（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いかなる操作も、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。
【００２３】
［0045］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、Ｈ．２６１、Ｈ．２６３又はＨ．２６４／ＡＶＣ）内の「ブロック」と呼ぶことができるか、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（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いかなる操作も、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も留意すべきである。
【００２４】
［0046］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何れの予測モード（例えば、イントラピクチャ予測又はインターピクチャ予測）を使用する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。
【００２５】
［0047］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。
【００２６】
［0048］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、符号器は、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／ＨＥＶＣ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／ＨＥＶＣ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は、異なるサイズ及び数を有し得る。

【００２７】
［0049］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。
【００２８】
［0050］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、ピクチャを処理のため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することができる。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するか又はネットワーク伝送内で失われる場合、コーデックは、破損するか又は失われたデータに依存することなく、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／ＨＥＶＣ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の様々なピクチャは、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も留意すべきである。
【００２９】
［0051］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。
【００３０】
［0052］図２Ａは、本開示の実施形態と合致する、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に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ピクチャ（「元のピクチャ」と呼ぶ）の組を含み得る。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、又は処理のため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を反復的な方法で実行することができ、符号器は、プロセス２００Ａの１回の反復において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。
【００３１】
［0053］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及び予測されたＢＰＵ２０８を生成することができる。符号器は、元のＢＰＵから、予測された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ＢＰＵ２１０を変換段階２１２及び量子化段階２１４にフィードして、量子化された変換係数２１６を生成することができる。符号器は、予測データ２０６及び量子化された変換係数２１６をバイナリコード化段階２２６にフィードして、映像ビットストリーム２２８を生成することができる。構成要素２０２、２０４、２０６、２０８、２１０、２１２、２１４、２１６、２２６及び２２８は、「順方向経路」と呼ぶことができる。プロセス２００Ａ中、符号器は、量子化段階２１４後、量子化された変換係数２１６を逆量子化段階２１８及び逆変換段階２２０にフィードして、再構築された残差ＢＰＵ２２２を生成することができる。符号器は、再構築された残差ＢＰＵ２２２を、予測されたＢＰＵ２０８に加え

て、プロセス２００Ａの次の反復の予測段階２０４に使用される予測基準２２４を生成することができる。プロセス２００Ａの構成要素２１８、２２０、２２２及び２２４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。

【００３２】

[0054] 符号器は、プロセス２００Ａ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ＢＰＵを符号化し、（再構築経路内で）元のピクチャの次の元のＢＰＵを符号化するための予測された基準２２４を生成することができる。元のピクチャの全ての元のＢＰＵを符号化した後、符号器は、映像シーケンス２０２内の次のピクチャの符号化に進むことができる。

【００３３】

[0055] プロセス２００Ａを参照すると、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２０２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方法の任意のアクションを指すことができる。

【００３４】

[0056] 予測段階２０４では、現在の反復において、符号器が元のＢＰＵ及び予測基準２２４を受信し、予測操作を行って予測データ２０６及び予測されたＢＰＵ２０８を生成することができる。予測基準２２４は、プロセス２００Ａ前の反復の再構築経路から生成され得る。予測段階２０４の目的は、予測データ２０６を抽出することにより、情報の冗長性を減らすことであり、予測データ２０６は、予測データ２０６及び予測基準２２４から予測されたＢＰＵ２０８として元のＢＰＵを再構築するために使用され得る。

【００３５】

[0057] 理想的には、予測されたＢＰＵ２０８は、元のＢＰＵ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、予測されたＢＰＵ２０８は、概して、元のＢＰＵ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、予測されたＢＰＵ２０８を生成した後、それを元のＢＰＵから減算して残差ＢＰＵ２１０を生成することができる。例えば、符号器は、予測されたＢＰＵ２０８のピクセルの値（例えば、グレースケール値又はＲＧＢ値）を元のＢＰＵ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ＢＰＵの対応するピクセルと、予測されたＢＰＵ２０８との間のかかる減算の結果、残差ＢＰＵ２１０の各ピクセルは、残差値を有し得る。元のＢＰＵと比較して、予測データ２０６及び残差ＢＰＵ２１０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く元のＢＰＵを再構築するためにそれらを使用することができる。

【００３６】

[0058] 残差ＢＰＵ２１０を更に圧縮するために、変換段階２１２において、符号器は、残差ＢＰＵ２１０を２次元「基底パターン」の組に分解することにより、残差ＢＰＵ２１０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ＢＰＵ２１０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ＢＰＵ２１０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。基底パターンの何れも、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）から再現することができない。換言すれば、分解は、残差ＢＰＵ２１０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、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、変換係数は、基底関数に関連する係数に類似する。

【００３７】

[0059] 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２１２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２１２における変換は、可逆的である。即ち、符号器は、変換の逆操作（「逆変換」と呼ぶ）によって残差ＢＰＵ２１０を復元することがで

きる。例えば、残差ＢＰＵ２１０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を、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、変換係数のみを記録することができ、復号器は、符号器から基底パターンを受信することなく、変換係数から残差ＢＰＵ２１０を再構築することができる。残差ＢＰＵ２１０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く残差ＢＰＵ２１０を再構築するために使用され得る。従って、残差ＢＰＵ２１０が更に圧縮される。

【００３８】

［0060］符号器は、量子化段階２１４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ことが得意であるため、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く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２１４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り、量子化された変換係数２１６を生成することができる。かかる操作後、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、ゼロ値の量子化された変換係数２１６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、量子化された変換係数２１６は、量子化の逆操作（「逆量子化」と呼ぶ）内で変換係数に再構築することができる。

【００３９】

［0061］符号器は、丸め操作内でかかる除算の剰余を無視するため、量子化段階２１４は、非可逆であり得る。典型的には、量子化段階２１４は、プロセス２００Ａ内で最大の情報損失に寄与し得る。情報損失が大きいほど、量子化された変換係数２１６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。

【００４０】

［0062］バイナリコード化段階２２６において、符号器は、例えば、エントロピーコード化、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２０６及び量子化された変換係数２１６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２０６及び量子化された変換係数２１６に加えて、符号器は、例えば、予測段階２０４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２１２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２２６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２２６の出力データを使用して映像ビットストリーム２２８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２２８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。

【００４１】

［0063］プロセス２００Ａの再構築経路を参照すると、逆量子化段階２１８では、符号器は、量子化された変換係数２１６に対して逆量子化を行って、再構築された変換係数を生成することができる。逆変換段階２２０では、符号器は、再構築された変換係数に基づいて、再構築された残差ＢＰＵ２２２を生成することができる。符号器は、再構築された残差ＢＰＵ２２２を、予測されたＢＰＵ２０８に加えて、プロセス２００Ａの次の反復内で使用される予測基準２２４を生成することができる。

【００４２】

［0064］映像シーケンス２０２を符号化するためにプロセス２００Ａの他のバリエーシ

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２００Ａ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２００Ａの１つ又は複数の段階を単一の段階に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２００Ａ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２１２と量子化段階２１４とを単一の段階に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２００Ａは、追加の段階を含み得る。一部の実施形態では、プロセス２００Ａは、図２Ａ内の１つ又は複数の段階を省くことができる。

【００４３】

[0065] 図２Ｂは、本開示の実施形態に合致する、符号化プロセスの別の例２００Ｂの概略図を示す。プロセス２００Ｂは、プロセス２００Ａから修正され得る。例えば、プロセス２００Ｂは、ハイブリッド映像コード化規格（例えば、Ｈ．２６ｘシリーズ）に準拠する符号器によって使用され得る。プロセス２００Ａと比較して、プロセス２００Ｂの順方向経路は、モード決定段階２３０を更に含み、予測段階２０４を空間的予測段階２０４２及び時間的予測段階２０４４に分ける。プロセス２００Ｂの再構築経路は、ループフィルタ段階２３２及びバッファ２３４を追加で含む。

【００４４】

[0066] 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ＢＰＵ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ＢＰＵのピクセルを使用することができる。即ち、空間的予測における予測基準２２４は、隣接ＢＰＵ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、インターピクチャ予測又は「インター予測」）は、現ＢＰＵ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。即ち、時間的予測における予測基準２２４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。

【００４５】

[0067] プロセス２００Ｂを参照すると、順方向経路において、符号器は、空間的予測段階２０４２及び時間的予測段階２０４４で予測操作を行う。例えば、空間的予測段階２０４２では、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ＢＰＵに関して、予測基準２２４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ＢＰＵを含み得る。符号器は、隣接ＢＰＵを外挿することにより、予測されたＢＰＵ２０８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、予測されたＢＰＵ２０８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、符号器が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ＢＰＵは、垂直方向（例えば、元のＢＰＵの上）、水平方向（例えば、元のＢＰＵの左）、対角線方向（例えば、元のＢＰＵ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ＢＰＵに対して位置し得る。イントラ予測では、予測データ２０６は、例えば、使用される隣接ＢＰＵの位置（例えば、座標）、使用される隣接ＢＰＵのサイズ、外挿のパラメータ、元のＢＰＵに対する使用される隣接ＢＰＵの方向等を含み得る。

【００４６】

[0068] 別の例では、時間的予測段階２０４４では、符号器は、インター予測を行うことができる。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、再構築された残差ＢＰＵ２２２を、予測されたＢＰＵ２０８に加えて、再構築された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されたＢＰＵが生成されると、符号器は、参照ピクチャとして再構築された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ため、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、符号器は、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。

【００４７】

［0069］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。インター予測では、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。

【００４８】

［0070］予測された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、予測された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、その中では、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加えることができる。

【００４９】

［0071］一部の実施形態では、インター予測は、単方向又は双方向であり得る。単方向のインター予測は、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ー予測ピクチャである。双方向のインター予測は、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ー予測ピクチャである。

【００５０】

［0072］プロセス２００Ｂの順方向経路を引き続き参照すると、空間的予測段階２０４２及び時間的予測段階２０４４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ー予測の１つ）を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された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、対応する予測されたＢＰＵ２０８及び予測されたデータ２０６を生成することができる。

【００５１】

[0073] 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、符号器は、後に使用するために（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ー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、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、インター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）を減らすか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。ループフィルタされた参照ピクチャは、後に使用するために（例えば、映像シーケンス２０２の将来のピクチャのためのインター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「復号されたピクチャバッファ」）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、量子化された変換係数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。

【００５２】

[0074] 図３Ａは、本開示の実施形態に合致する、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は、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に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と非常に類似し得る。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を反復的な方法で実行することができ、復号器は、プロセス３００Ａの１回の反復において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。

【００５３】

[0075] 図３Ａでは、復号器は、符号化されたピクチャの基本処理単位（「符号化された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では、復号器は、その部分を予測データ２０６及び量子化された変換係数２１６に復号することができる。復号器は、量子化された変換係数２１６を逆量子化段階２１８及び逆変換段階２２０にフィードして、再構築された残差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、予測されたＢＰＵ２０８を生成することができる。復号器は、再構築された残差ＢＰＵ２２２を、予測されたＢＰＵ２０８に加えて、予測された基準２２４を生成することができる。一部の実施形態では、予測された基準２２４がバッファ（例えば、コンピュータメモリ内の復号されたピクチャバッファ）内に記憶され得る。復号器は、プロセス３００Ａの次の反復内で予測操作を行うための予測された基準２２４を予測段階２０４にフィードすることができる。

【００５４】

[0076] 復号器は、プロセス３００Ａを反復的に実行して、符号化されたピクチャの各符号化されたＢＰＵを復号し、符号化されたピクチャの次の符号化されたＢＰＵを符号化

するための予測された基準２２４を生成することができる。符号化されたピクチャの全ての符号化された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内の次の符号化されたピクチャの復号に進むことができる。

【００５５】

［0077］ バイナリ復号段階３０２では、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、エントロピーコード化、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及び量子化された変換係数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がネットワーク上においてパケット単位で伝送される場合、復号器は、映像ビットストリーム２２８をパケット化解除してからそれをバイナリ復号段階３０２にフィードすることができる。

【００５６】

［0078］ 図３Ｂは、本開示の実施形態に合致する、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と比較して、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。

【００５７】

［0079］ プロセス３００Ｂでは、復号されている符号化されたピクチャ（「現ピクチャ」と呼ぶ）の符号化された基本処理単位（「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何れの予測モードが符号器によって使用された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ー予測が符号器によって使用された場合、予測データ２０６は、インター予測、インター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。インター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。

【００５８】

［0080］ 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）を行うか、又は時間的予測段階２０４４で時間的予測（例えば、インター予測）を行う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に示されており、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、予測された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、予測されたＢＰＵ２０８と、再構築された残差ＢＰＵ２２２とを加えて、予測基準２２４を生成することができる。

【００５９】

［0081］ プロセス３００Ｂでは、復号器は、プロセス３００Ｂの次の反復内で予測操作を行うための予測された基準２２４を空間的予測段階２０４２又は時間的予測段階２０４４にフィードすることができる。例えば、現ＢＰＵが空間的予測段階２０４２においてイ

ントラ予測を使用して復号される場合、予測基準２２４（例えば、復号された現ＢＰＵ）を生成した後、復号器は、後に使用するために（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。現ＢＰＵが時間的予測段階２０４４においてインター予測を使用して復号される場合、予測基準２２４（例えば、全てのＢＰＵが復号されている参照ピクチャ）を生成した後、符号器は、ループフィルタ段階２３２に予測基準２２４をフィードして歪み（例えば、ブロッキングアーティファクト）を減らすか又はなくすことができる。復号器は、図２Ｂに記載した方法で予測基準２２４にループフィルタを適用することができる。ループフィルタされた参照ピクチャは、後に使用するために（例えば、映像ビットストリーム２２８の将来の符号化ピクチャのためのインター予測参照ピクチャとして使用するために）バッファ２３４（例えば、コンピュータメモリ内の復号されたピクチャバッファ）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、現ＢＰＵを符号化するためにインター予測が使用されたことを予測データ２０６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。

【００６０】

［0082］ 図４は、本開示の実施形態に合致する、映像を符号化又は復号するための機器４００の一例のブロック図である。図４に示すように、機器４００は、プロセッサ４０２を含み得る。プロセッサ４０２が本明細書に記載の命令を実行するとき、機器４００は、映像を符号化又は復号するための専用マシンになり得る。プロセッサ４０２は、情報を操作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４０２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、知的財産（ＩＰ）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４０２は、単一の論理構成要素としてグループ化されるプロセッサの組であり得る。例えば、図４に示すように、プロセッサ４０２は、プロセッサ４０２ａ、プロセッサ４０２ｂ及びプロセッサ４０２ｎを含む複数のプロセッサを含み得る。

【００６１】

［0083］ 機器４００は、データ（例えば、命令、コンピュータコード、中間データ等の組）を記憶するように構成されるメモリ４０４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２００Ａ、２００Ｂ、３００Ａ又は３００Ｂ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２０２、映像ビットストリーム２２８又は映像ストリーム３０４）を含み得る。プロセッサ４０２は、プログラム命令及び処理用データに（例えば、バス４１０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４０４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４０４は、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）メモリ群でもあり得る。

【００６２】

［0084］ 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。

【００６３】

［0085］曖昧さを招くことなく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であり得るか、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され得る。

【００６４】

［0086］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は、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。

【００６５】

［0087］一部の実施形態では、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を任意選択的に機器４００が更に含み得る。図４に示すように、周辺装置は、これのみに限定されないが、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。

【００６６】

［0088］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は、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は、専用データ処理回路（例えば、ＦＰＧＡ、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。

【００６７】

［0089］ＶＶＣでは、ピクチャが１つ又は複数のタイル行及び１つ又は複数のタイル列に分けられる。タイルはピクチャの矩形領域を覆う一連のＣＴＵである。タイル内のＣＴＵは、そのタイル内のラスタ走査順序によって走査される。スライスは、完全なタイルの整数個又はピクチャのタイル内の連続した完全なＣＴＵ行の整数個で構成される。その結果、それぞれの垂直スライス境界は垂直タイル境界でもある。スライスの水平境界は、タイル境界ではないが、タイル内の水平ＣＴＵ境界で構成されることがあり得る。これは、タイル内の連続した完全なＣＴＵ行の整数個でそれぞれ構成される複数の矩形スライスへとタイルを分割するときに生じる。

【００６８】

［0090］ＶＶＣでは、ネットワーク抽象化層（ＮＡＬ）単位形式の一連のビットであるコード化映像ビットストリーム又はバイトストリームが１つ又は複数のコード化映像シーケンス（ＣＶＳ）を形成し、各ＣＶＳは１つ又は複数のＣＬＶＳ（coded layer video sequence）で構成される。ＣＬＶＳは一連のピクチャ単位（ＰＵ）であり、各ＰＵは厳密に１つのコード化ピクチャを含む。

【００６９】

[0091] ＰＵは、ペイロードとしてのピクチャヘッダ構文構造、１つ又は複数の映像コード化層（ＶＣＬ）ＮＡＬ単位を含む１つのコード化ピクチャ、及びゼロ以上の他の非ＶＣＬ　ＮＡＬ単位を含む、ゼロ又は１つのピクチャヘッダ（ＰＨ）ＮＡＬ単位で構成される。ＶＣＬ　ＮＡＬ単位は、スライスヘッダ及びスライスデータで構成されるコード化スライスを含む。

【００７０】

[0092] ＶＶＣでは量子化パラメータ（ＱＰ）の範囲を０～６３とすることができ、初期ＱＰのシグナリングをしかるべく変更することができる。slice_qp_deltaの非ゼロ値がスライスヘッダ内にコード化される場合、SliceQpYの初期値がスライスレベルで修正される。とりわけ、init_qp_minus26の値は（－２６＋QpBdOffsetY）～＋３７の範囲内にあるように修正される。変換ブロックのサイズが４の累乗ではない場合、変換プロセスによる暗示的なスケーリングを補償するために、１８１／２５６（又は１８１／１２８）による乗算によってではなく、ＱＰ又はQP levelScaleテーブルに対する修正と共に変換係数が処理される。変換スキップブロックでは、最小許容ＱＰが４として定められており、その理由はＱＰが４に等しいとき量子化ステップサイズが１になるからである。

【００７１】

[0093] 加えて、ＱＰ値はＣＵごとに又は量子化グループごとに変更することができる。ルマ成分及びクロマ成分のためのデルタＱＰ値は別々にシグナリングすることができる。

【００７２】

[0094] ルマコード化ブロックごとに、変数$qP_{Y_PREV}$を以下のようにまず導出する：

－ 下記の条件の１つ又は複数が真である場合$qP_{Y_PREV}$をSliceQpYに等しく設定する：

－ 現在の量子化グループがスライス内の第１の量子化グループである。

－ 現在の量子化グループがタイル内の第１の量子化グループである。

－ さもなければ、$qP_{Y_PREV}$は、復号順における前の量子化グループ内の最後のルマコード化単位のルマ量子化パラメータ$Qp_Y$に等しく設定する。

【００７３】

[0095] 第２に、変数$qP_{Y_A}$を以下のように導出する：

－ 下記の条件の１つ又は複数が真である場合$qP_{Y_A}$を$qP_{Y_PREV}$に等しく設定する：

－ 現在の量子化グループの左の隣接ブロックを利用することができない。

－ 現在の量子化グループの左の隣接ブロック及び現在のコード化ブロックが異なるコード化ツリーブロック（ＣＴＢ）内にある。

－ さもなければ、$qP_{Y_A}$は現在の量子化グループの上側のコード化単位のルマ量子化パラメータに等しく設定する。

【００７４】

[0096] 第３に、変数$qP_{Y_B}$を以下のように導出する：

－ 下記の条件の１つ又は複数が真である場合$qP_{Y_B}$を$qP_{Y_PREV}$に等しく設定する：

－ 現在の量子化グループの上の隣接ブロックを利用することができない。

－ 現在の量子化グループの上の隣接ブロック及び現在のコード化ブロックが異なるコード化ツリーブロック（ＣＴＢ）内にある。

－ さもなければ、$qP_{Y_B}$は現在の量子化グループの左側のコード化単位のルマ量子化パラメータに等しく設定する。

【００７５】

[0097] 第４に、現在の量子化グループがブロック内のコード化ツリーブロック（ＣＴＢ）行内の第１の量子化グループであり、現在の量子化グループの上の隣接ブロックを利用することができる場合、

qPY_PREDをqPY_Bに設定し、

さもなければ

qPY_PRED=(qPY_A+qPY_B+1)〉〉1

【００７６】

[0098] qPY_PREDを導出した後、以下の等式１を使用して現在のルマコード化ブロッ

クの量子化パラメータQp′Yを導出することができる：

Qp′Y=((qPY_PRED+CuQpDeltaVal+64+2*QpBdOffsetY)%(64+QpBdOffsetY)) （等式１）

但し、QpBdOffsetYは6*sps_bitdepth_minus8に等しく、変数CuQpDeltaValはルマコード化ブロックの量子化パラメータとその予測値との差を指定する。

【００７７】

[0099] ＶＶＣでは、CuQpDeltaValがcu_qp_delta_abs*(1-2*cu_qp_delta_sign_flag)として指定され、但し、cu_qp_delta_abs及びcu_qp_delta_sign_flagはＣＵレベルにおいてビットストリーム内でシグナリングされる構文要素である。cu_qp_delta_abs及びcu_qp_delta_sign_flagがビットストリーム内にない場合、CuQpDeltaValは０であると推論することができる。

【００７８】

[0100] クロマコード化ブロックのための量子化パラメータは$Qp_Y$と異なり得る。クロマ量子化パラメータ（$Qp_{Cb}$、$Qp_{Cr}$、$Qp_{CbCr}$）とルマ量子化パラメータとの間のオフセットはビットストリーム内でシグナリングされ得る。ＶＶＣでは、以下の等式２～４を使用してクロマ量子化パラメータQp′Cb及びQp′Cr、並びにジョイントＣｂ－Ｃｒコード化のためのＱＰであるQp′CbCrを導出することができ：

Qp′Cb=Clip3(-QpBdOffset$_C$,63,$qP_{Cb}$+pps_cb_qp_offset+slice_cb_qp_offset+CuQpOffset$_{Cb}$)+QpBdOffset$_C$ （等式２）

Qp′Cr=Clip3(-QpBdOffset$_C$,63,$qP_{Cr}$+pps_cr_qp_offset+slice_cr_qp_offset+CuQpOffset$_{Cr}$)+QpBdOffset$_C$ （等式３）

Qp′CbCr=Clip3(-QpBdOffset$_C$,63,$qP_{CbCr}$+pps_cbcr_qp_offset+slice_cbcr_qp_offset+CuQpOffset$_{CbCr}$)+QpBdOffset$_C$ （等式４）

但し、$qP_{Cb}$、$qP_{Cr}$、及び$qP_{CbCr}$は等式５～８を使用して$Qp_Y$のクリッピングされた値の入力を用いて、参照表から導出することができる：

$qPi_{Chroma}$=Clip3(-QpBdOffset,63,$Qp_Y$-QpBdOffset) （等式５）

$qP_{Cb}$=ChromaQpTable[0][$qP_{Chroma}$] （等式６）

$qP_{Cr}$=ChromaQpTable[1][$qP_{Chroma}$] （等式７）

$qP_{CbCr}$=ChromaQpTable[2][$qP_{Chroma}$] （等式８）

【００７９】

[0101] cu_chroma_qp_offset_flagが０に等しい場合、CuQpOffset$_{Cb}$、CuQpOffset$_{Cr}$、及びCuQpOffset$_{CbCr}$は０に設定され、cu_chroma_qp_offset_flagが１に等しい場合は等式９～１１を使用して導出することができる：

CuQpOffset$_{Cb}$=cb_qp_offset_list[cu_chroma_qp_offset_idx] （等式９）

CuQpOffset$_{Cr}$=cr_qp_offset_list[cu_chroma_qp_offset_idx] （等式１０）

CuQpOffset$_{CbCr}$=joint_cbcr_qp_offset_list[cu_chroma_qp_offset_idx] （等式１１）

但しcu_chroma_qp_offset_flag及びcu_chroma_qp_offset_idxはビットストリーム内でシグナリングされる構文要素である。

【００８０】

[0102] 上記で論じたように、ＱＰの導出に使用可能なCuQpDeltaValを導出するためにcu_qp_delta_abs及びcu_qp_delta_sign_flagがシグナリングされる。クロマＱＰの導出に使用可能なCuQpOffset$_{Cb}$、CuQpOffset$_{Cr}$、及びCuQpOffset$_{CbCr}$を導出するために、cu_chroma_qp_offset_flag、cu_chroma_qp_offset_idx、cb_qp_offset_list[i]、cr_qp_offset_list[i]、及びjoint_cbcr_qp_offset_list[i]がシグナリングされる。

【００８１】

[0103] 以下は関係する構文のシグナリングプロセスの導入である。まず、ＣＵデルタＱＰのための例示的なＰＰＳ構文を示す図５に示すように、cu_qp_delta_enabled_flag、cu_qp_delta_subdiv、cu_chroma_qp_offset_enabled_flag、及びcu_chroma_qp_offset_subdivをピクチャパラメータセット（ＰＰＳ）内でシグナリングすることができ

る。
【００８２】
［0104］その後、ＣＵデルタＱＰのための例示的なコード化ツリー構文を示す図６に示すように、変数IsCuQpDeltaCoded及びIsCuChromaQpOffsetCoded、量子化パラメータグループの位置、並びに変数qgOnY及びqgOnCをコード化ツリーレベルにおいて導出することができる。
【００８３】
［0105］更に、図７に示すように、ＣＵデルタＱＰのための例示的な変換単位レベル構文を示す、IsCuQpDeltaCoded及びIsCuChromaQpOffsetCodedがコード化単位レベルにおいて導出されていることを条件とし、cu_qp_delta_abs/cu_qp_delta_sign_flag及びcu_chroma_qp_offset_flag/cu_chroma_qp_offset_idxが変換単位においてシグナリングされる。
【００８４】
［0106］図５の例では、cu_qp_delta_subdivはcu_qp_delta_abs及びcu_qp_delta_sign_flagを伝えるコード化単位の最大cbSubdiv値を指定し、cu_chroma_qp_offset_subdivはcu_chroma_qp_offset_flagを伝えるコード化単位の最大cbSubdiv値を指定する。cbSubdivは、その値がコード化単位のサイズに関係する変数である。より小さいコード化単位はより大きいcbSubdivの値を有する。コード化単位を複数の副コード化単位に分割することにより、cbSubdivの値が増加する。cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivの値の範囲は、スライスレベル及びスライスの種類に基づいて導出されるMaxMttDepthYと呼ばれる変数に依存する。

MaxMttDepthY=slice_max_mtt_hierarchy_depth_luma　（等式１２）

但し、slice_max_mtt_hierarchy_depth_lumaは、例示的なスライスヘッダ構文を示す図８に示すように、スライスヘッダ内でシグナリングされる。
【００８５】
［0107］上記で説明したように、cu_qp_delta_abs/cu_qp_delta_sign_flag及びcu_chroma_qp_offset_flagを伝え得るコード化単位の最大深度を決定するために、２つの構文要素cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivがＰＰＳレベル内でシグナリングされる。但しcu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivの値の範囲は、スライスレベル及びスライスの種類に基づいて導出される変数MaxMttDepthYに依存する。従って、ＰＰＳレベルの構文要素はスライスレベル構文に依存する。
【００８６】
［0108］ビットストリーム構文構造において、ＰＰＳはスライスレベルよりも高レベルにあり、ＰＰＳの構文はスライス構文の前に来る。復号器では、低レベル構文を構文解析するとき高レベル構文の値を参照することができる。しかし高レベル構文を構文解析するとき、低レベル構文の値を参照することはできない。従って現在のＶＶＣ技法では、cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivがスライスヘッダ構文に依存することは、解決される必要がある論理的問題を引き起こす。
【００８７】
［0109］上記の問題に対処するために、本開示の様々な実施形態において解決策を提供する。一部の実施形態では、cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivはslice_max_mtt_hierarchy_depth_lumaがシグナリングされた後でスライスヘッダに移すことができる。それにより、cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivはもはやＰＰＳレベルの構文要素ではない。スライスヘッダ構文の一例を図９（例えば要素９０１）に示す。
【００８８】
［0110］図９に示す例では、cu_qp_delta_enabled_flag及びcu_chroma_qp_offset_enabled_flagがＰＰＳ内でシグナリングされる。一部の実施形態では、図１０（例えば要素１００１）に示すようにcu_qp_delta_enabled_flag及びcu_chroma_qp_offset_enabled_flagがスライスヘッダ内でシグナリングされ得る。

【００８９】
［0111］図１０に示す例では、cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivの範囲を以下のように決定することができる。例えばcu_qp_delta_subdivの値の範囲は以下のように指定することができる。slice_typeがIに等しい場合、cu_qp_delta_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeIntraY+MaxMttDepthY)の範囲内にある（両端を含む）。そうではない（slice_typeがIに等しくない）場合、cu_qp_delta_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeInterY+MaxMttDepthY)の範囲内にある。存在しない場合、cu_qp_delta_subdivの値は０に等しいと推論することができる。
【００９０】
［0112］cu_chroma_qp_offset_subdivの値の範囲は以下のように指定することができる。slice_typeがIに等しい場合、cu_chroma_qp_offset_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeIntraY+MaxMttDepthY)の範囲内にある。そうではない（slice_typeがIに等しくない）場合、cu_chroma_qp_offset_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeInterY+MaxMttDepthY)の範囲内にある。存在しない場合、cu_chroma_qp_offset_subdivの値は０に等しいと推論することができる。
【００９１】
［0113］一部の実施形態では、cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivがピクチャヘッダに移され、それと同時にMaxMttDepthYを導出するために使用される構文要素もピクチャヘッダに移され、そのためcu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivの範囲はスライスレベル構文に依存しない。
【００９２】
［0114］１つのピクチャがインター及びイントラの異なるスライスの種類を有する複数のスライスを含み得るので。従ってこの実施形態では、cu_qp_delta_subdivは２つの構文要素、つまりph_cu_qp_delta_subdiv_intra_slice及びph_cu_qp_delta_subdiv_inter_sliceに分けられる。cu_chroma_qp_offset_subdivは２つの構文要素、つまりph_cu_chroma_qp_offset_subdiv_intra_slice及びph_cu_chroma_qp_offset_subdiv_inter_sliceに分けられる。ph_cu_qp_delta_subdiv_intra_slice及びph_cu_chroma_qp_offset_subdiv_intra_sliceは現ピクチャ内のイントラスライス用であり、ph_cu_qp_delta_subdiv_inter_slice及びph_cu_chroma_qp_offset_subdiv_inter_sliceは現ピクチャ内のインタスライス用である。同様に、２つの構文要素、つまりph_max_mtt_hierarchy_depth_intra_slice_luma及びph_max_mtt_hierarchy_depth_inter_sliceがイントラスライス及びインタスライスのMaxMttDepthYのためにシグナリングされる。
【００９３】
［0115］ピクチャヘッダ構文の一例を図１１の表１１に示す。表１１に示すように、ph_cu_qp_delta_subdiv_intra_slice（例えば要素１１０１）、ph_cu_chroma_qp_offset_subdiv_intra_slice（例えば要素１１０２）、ph_cu_qp_delta_subdiv_inter_slice（例えば要素１１０３）、及びph_cu_chroma_qp_offset_subdiv_inter_slice（例えば要素１１０４）をイタリック体及び灰色で示す。
【００９４】
［0116］イントラスライスに関して、ph_cu_qp_delta_subdiv_intra_sliceはcu_qp_delta_abs及びcu_qp_delta_sign_flagを伝えるイントラスライス内のコード化単位の最大cbSubdiv値を指定する。ph_cu_qp_delta_subdiv_intra_sliceの値は０～2*(CtbLog2SizeY-MinQtLog2SizeIntraY+ph_max_mtt_hierarchy_depth_intra_slice_luma)の範囲内にある。存在しない場合、ph_cu_qp_delta_subdiv_intra_sliceの値は０に等しいと推論することができる。
【００９５】
［0117］ph_cu_chroma_qp_offset_subdiv_intra_sliceはcu_chroma_qp_offset_flagを伝えるイントラスライス内のコード化単位の最大cbSubdiv値を指定する。ph_cu

_chroma_qp_offset_subdiv_intra_sliceの値は０～2*(CtbLog2SizeY-MinQtLog2SizeIntraY+ph_max_mtt_hierarchy_depth_intra_slice_luma)の範囲内にある。存在しない場合、ph_cu_chroma_qp_offset_subdiv_intra_sliceの値は０に等しいと推論することができる。

【００９６】

[0118] 開示する実施形態では、ph_max_mtt_hierarchy_depth_intra_slice_lumaがピクチャヘッダ内でシグナリングされ、「I」に等しいsh_slice_typeを有するスライス（即ちイントラ予測スライス）内の４分木の葉のマルチタイプツリー分割から生じるコード化単位の最大階層深度を指定する。CtbLog2SizeY及びMinQtLog2SizeIntraYは以下の等式１３～１５を使用して導出され、これらの等式ではCtbLog2SizYは「I」に等しいslice_typeを有するスライス（即ちイントラ予測スライス）内のコード化ツリー単位のルマコード化ツリーブロックのサイズを表し、MinQtLog2SizeIntraYは「I」に等しいslice_typeを有するスライス内のコード化ツリー単位の４分木分割から生じるルマリーフブロックのルマサンプル内の最小サイズを表す。

CtbLog2SizeY=sps_log2_ctu_size_minus5+5 （等式１３）

MinQtLog2SizeIntraY=sps_log2_diff_min_qt_min_cb_intra_slice_luma+MinCbLog2SizeY （等式１４）

MinCbLog2SizeY=sps_log2_min_luma_coding_block_size_minus2+2 （等式１５）

【００９７】

[0119] sps_log2_ctu_size_minus5、sps_log2_diff_min_qt_min_cb_intra_slice_luma、及びsps_log2_min_luma_coding_block_size_minus2はＳＰＳ内でシグナリングされる構文要素である。

【００９８】

[0120] 変数CuQpDeltaSubdivは、cu_qp_delta_abs及びcu_qp_delta_sign_flagを伝えるコード化単位の最大cbSubdiv値として導出され、変数CuChromaQpOffsetSubdivはcu_chroma_qp_offset_flagを伝えるコード化単位の最大cbSubdiv値として導出される。これらの２つの変数は等式１６及び等式１７としてそれぞれ導出される。

CuQpDeltaSubdiv=ph_cu_qp_delta_subdiv_intra_slice （等式１６）

CuChromaQpOffsetSubdiv=ph_cu_chroma_qp_offset_subdiv_intra_slice （等式１７）

【００９９】

[0121] インタスライスに関して、ph_cu_qp_delta_subdiv_inter_sliceはインタスライス内でcu_qp_delta_abs及びcu_qp_delta_sign_flagを伝えるコード化単位の最大cbSubdiv値を指定する。ph_cu_qp_delta_subdiv_inter_sliceの値は０～2*(CtbLog2SizeY-MinQtLog2SizeInterY+ph_max_mtt_hierarchy_depth_inter_slice)の範囲内にあり得る。存在しない場合、ph_cu_qp_delta_subdiv_inter_sliceの値は０に等しいと推論することができる。ph_cu_chroma_qp_offset_subdiv_inter_sliceはcu_chroma_qp_offset_flagを伝えるインタスライス内のコード化単位の最大cbSubdiv値を指定する。ph_cu_chroma_qp_offset_subdiv_inter_sliceの値は０～2*(CtbLog2SizeY-MinQtLog2SizeInterY+ph_max_mtt_hierarchy_depth_inter_slice)の範囲内にある。存在しない場合、ph_cu_chroma_qp_offset_subdiv_inter_sliceの値は０に等しいと推論することができる。

【０１００】

[0122] ph_max_mtt_hierarchy_depth_inter_sliceはピクチャヘッダ内でシグナリングすることができ、「I」に等しくないsh_slice_typeを有するスライス（即ち「P」又は「B」に等しいslice_typeを有するインター予測スライス）内の４分木の葉のマルチタイプツリー分割から生じるコード化単位の最大階層深度を指定する。CtbLog2SizY及びMinQtLog2SizeInterYは以下の等式１８～２０を使用して導出され、これらの等式ではCtbLog2SizYは「I」に等しくないslice_typeを有するスライス（即ち「P」又は「B」に等

しいslice_typeを有するインター予測スライス）内のコード化ツリー単位のルマコード化ツリーブロックのサイズを表し、MinQtLog2SizeInterYは「I」に等しくないslice_typeを有するスライス内のコード化ツリー単位の４分木分割から生じるルマリーフブロックのルマサンプル内の最小サイズを表す。

CtbLog2SizeY=sps_log2_ctu_size_minus5+5　（等式１８）

MinQtLog2SizeInterY=sps_log2_diff_min_qt_min_cb_inter_slice_luma+MinCbLog2SizeY　（等式１９）

MinCbLog2SizeY=sps_log2_min_luma_coding_block_size_minus2+2　（等式２０）

【０１０１】

[0123] sps_log2_ctu_size_minus5、sps_log2_diff_min_qt_min_cb_inter_slice_luma、及びsps_log2_min_luma_coding_block_size_minus2はＳＰＳ内でシグナリングされる構文要素である。

【０１０２】

[0124] 変数CuQpDeltaSubdivは、cu_qp_delta_abs及びcu_qp_delta_sign_flagを伝えるコード化単位の最大cbSubdiv値として導出され、変数CuChromaQpOffsetSubdiはcu_chroma_qp_offset_flagを伝えるコード化単位の最大cbSubdiv値として導出される。これらの２つの変数は等式２１及び等式２２としてそれぞれ導出される。

CuQpDeltaSubdiv=ph_cu_qp_delta_subdiv_inter_slice　（等式２１）

CuChromaQpOffsetSubdiv=ph_cu_chroma_qp_offset_subdiv_inter_slice　（等式２２）

【０１０３】

[0125] 一部の実施形態では、cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivがどちらもＰＰＳレベルにおいて及びスライスヘッダ内でシグナリングされ得る。例えば図１２（例えば要素１２０１及び１２０２）に示すように、ＰＰＳ構文内でpps_cu_qp_delta_subdiv及びpps_cu_chroma_qp_offset_subdivがシグナリングされる。図１３（例えば要素１３０１）に示すように、スライスヘッダ内でslice_cu_qp_delta_subdiv及びslice_cu_chroma_qp_offset_subdivもシグナリングされる。

【０１０４】

[0126] 一部の実施形態では、以下の例で示すようにpps_cu_qp_delta_subdiv及びpps_cu_chroma_qp_offset_subdivの範囲がシーケンスパラメータセット（ＳＰＳ）の構文に依存する。この例ではpps_cu_qp_delta_subdivの値の範囲が次のように指定される：pps_cu_qp_delta_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeY+SpsMaxMttDepthY)の範囲内にある。存在しない場合、pps_cu_qp_delta_subdivの値は０に等しいと推論することができる。pps_cu_chroma_qp_offset_subdivの値の範囲は次のように指定される：pps_cu_chroma_qp_offset_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeY+SpsMaxMttDepthY)の範囲内にあり得る。存在しない場合、pps_cu_chroma_qp_offset_subdivの値は０に等しいと推論することができる。

【０１０５】

[0127] ctbLog2SizeYが定められる場合、MinQtLog2SizeY及びSpsMaxMttDepthYは以下のように導出することができる。

【０１０６】

[0128] 或るやり方では、MinQtLog2SizeYは：

min(MinQtLog2SizeIntraY, MinQtLog2SizeInterY)

又は

max(MinQtLog2SizeIntraY, MinQtLog2SizeInterY)

として導出することができる。

【０１０７】

[0129] MinQtLog2SizeIntraY及びMinQtLog2SizeIntraYは、ＶＶＣドラフト６に定められているような様々な技法を使用して導出できることが理解されよう。

【０１０８】
［0130］代替的なやり方で、MinQtLog2SizeYの値は以下の等式２３に基づいて導出することができる：
MinQtLog2SizeY=sps_log2_diff_min_qt_min_cb_luma+MinCbLog2SizeY　（等式２３）
【０１０９】
［0131］但しsps_log2_diff_min_qt_min_cb_lumaは図１４（例えば要素１４０１）に示すようにＳＰＳ内でシグナリングされる。MinCbLog2SizeYは、ＶＶＣドラフト６に定められているような様々な技法を使用して導出できることが理解されよう。
【０１１０】
［0132］SpsMaxMttDepthに関して、或るやり方ではSpsMaxMttDepthYを以下のように導出することができる：
min(sps_max_mtt_hierarchy_depth_intra_slice_luma, sps_max_mtt_hierarchy_depth_inter_slice)
又は
max(sps_max_mtt_hierarchy_depth_intra_slice_luma, sps_max_mtt_hierarchy_depth_inter_slice)
但しsps_max_mtt_hierarchy_depth_intra_slice_luma及びsps_max_mtt_hierarchy_depth_inter_sliceはＳＰＳ内でシグナリングされ得る。
【０１１１】
［0133］代替的なやり方で、SpsMaxMttDepthYの値は以下のように導出することができる：
SpsMaxMttDepthY=sps_max_mtt_depth_luma　（等式２４）
但しsps_max_mtt_depth_lumaは図１４（例えば要素１４０２）に示すようにＳＰＳ内でシグナリングされ得る。
【０１１２】
［0134］上記の例では、ＰＰＳ構文要素pps_cu_qp_delta_subdiv及びpps_cu_chroma_qp_offset_subdivがＳＰＳ構文に依存する。そのようなＰＰＳとＳＰＳとの間の構文解析の依存関係は望ましくない場合がある。この依存関係の問題に対処するために、一部の実施形態ではpps_cu_qp_delta_subdivの値の範囲を以下のように指定することができる。
【０１１３】
［0135］pps_cu_qp_delta_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeY+ppsMaxMttDepthY)の範囲内にある。存在しない場合、pps_cu_qp_delta_subdivの値は０に等しいと推論することができる。
【０１１４】
［0136］pps_cu_chroma_qp_offset_subdivの値の範囲は以下のように指定することができる。pps_cu_chroma_qp_offset_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeY+ppsMaxMttDepthY)の範囲内にある。存在しない場合、pps_cu_chroma_qp_offset_subdivの値は０に等しいと推論することができる。
【０１１５】
［0137］CtbLog2SizeY、MinQtLog2SizeY、及びppsMaxMttDepthYは以下のように導出される：
CtbLog2SizeY=pps_log2_ctb_size　（等式２５）
MinQtLog2SizeY=pps_log2_min_qt　（等式２６）
ppsMaxMttDepthY=pps_max_mtt_depth_luma　（等式２７）
【０１１６】
［0138］pps_log2_ctb_size、pps_log2_min_qt、及びpps_max_mtt_depth_lumaは図１５（例えば要素１５０１）内に示すようにＰＰＳ内でシグナリングされ得る。
【０１１７】

[0139] 上記の例では、slice_cu_qp_delta_subdiv及びslice_cu_chroma_qp_offset_subdivの範囲がスライスヘッダの構文に依存する。例えばslice_cu_qp_delta_subdivの値の範囲は以下のように指定することができる。slice_typeがIに等しい場合、slice_cu_qp_delta_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeIntraY+SliceMaxMttDepthY)の範囲内にある。そうではない（slice_typeがIに等しくない）場合、slice_cu_qp_delta_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeInterY+SliceMaxMttDepthY)の範囲内にある。存在しない場合、slice_cu_qp_delta_subdivの値は０又はpps_cu_qp_delta_subdivに等しいと推論することができる。

【０１１８】

[0140] slice_cu_chroma_qp_offset_subdivの値の範囲は以下のように指定することができる。slice_typeがIに等しい場合、slice_cu_chroma_qp_offset_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeIntraY+SliceMaxMttDepthY)の範囲内にある。そうではない（slice_typeがIに等しくない）場合、slice_cu_chroma_qp_offset_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeInterY+SliceMaxMttDepthY)の範囲内にある。存在しない場合、slice_cu_chroma_qp_offset_subdivの値は０又はpps_cu_chroma_qp_offset_subdivに等しいと推論することができる。

【０１１９】

[0141] CtbLog2SizeY、MinQtLog2SizeIntraY、及びMinQtLog2SizeInterYが定められる場合、SliceMaxMttDepthYは

SliceMaxMttDepthY=slice_max_mtt_hierarchy_depth_luma （等式２８）

として導出することができる。

【０１２０】

[0142] slice_max_mtt_hierarchy_depth_lumaはスライスヘッダ内でシグナリングされ得る。

【０１２１】

[0143] 上記の例では、cu_qp_delta_subdivはslice_cu_qp_delta_subdivだと推論することができる。或いはcu_qp_delta_subdivはpps_cu_qp_delta_subdivだと最初に推論することができ、slice_cu_qp_delta_subdivが存在する場合、slice_cu_qp_delta_subdivがオーバライドし、cu_qp_delta_subdivはslice_cu_qp_delta_subdivだと推論することができる。cu_qp_delta_subdivの値は$Qp_Y$を導出するために使用することができる。

【０１２２】

[0144] 更に、cu_chroma_qp_offset_subdivはslice_cu_chroma_qp_offset_subdivだと推論することができる。或いはcu_chroma_qp_offset_subdivはpps_cu_chroma_qp_offset_subdivだと最初に推論することができ、slice_cu_chroma_qp_offset_subdivが存在する場合、slice_cu_chroma_qp_offset_subdivがオーバライドし、cu_chroma_qp_offset_subdivはslice_cu_chroma_qp_offset_subdivだと推論することができる。cu_chroma_qp_offset_subdivの値は$Qp_{Cb}$、$Qp_{Cr}$、$QP_{CbCr}$を導出するために使用することができる。

【０１２３】

[0145] 一部の実施形態では、cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivがどちらもＳＰＳレベルにおいて及びスライスヘッダ内でシグナリングされ得る。図１６（例えば要素１６０１）内に示すように、ＳＰＳ内でsps_cu_qp_delta_subdiv及びsps_cu_chroma_qp_offset_subdivをシグナリングすることができ、図１７（例えば要素１７０１）に示すように、スライスヘッダ内でslice_cu_qp_delta_subdiv及びslice_cu_chroma_qp_offset_subdivがシグナリングされる。

【０１２４】

[0146] 一部の実施形態では、sps_cu_qp_delta_subdiv及びsps_cu_chroma_qp_offset_subdivの範囲がＳＰＳの構文に依存する。例えばsps_cu_qp_delta_subdivの値の範囲は以下のように指定することができる。sps_cu_qp_delta_subdivの値は０～2*(

CtbLog2SizeY-MinQtLog2SizeY+SpsMaxMttDepthY)の範囲内にある。存在しない場合、sps_cu_qp_delta_subdivの値は０に等しいと推論することができる。sps_cu_chroma_qp_offset_subdivの値の範囲は以下のように指定される。sps_cu_chroma_qp_offset_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeY+SpsMaxMttDepthY)の範囲内にある。存在しない場合、sps_cu_chroma_qp_offset_subdivの値は０に等しいと推論することができる。

【０１２５】

[0147] ctbLog2SizeYが定められる場合、MinQtLog2SizeY及びSpsMaxMttDepthYを以下のように導出することができる。

【０１２６】

[0148] 或るやり方では、MinQtLog2SizeYは
min(MinQtLog2SizeIntraY, MinQtLog2SizeInterY)
又は
max(MinQtLog2SizeIntraY, MinQtLog2SizeInterY)
として導出することができる。

【０１２７】

[0149] 但しMinQtLog2SizeIntraY及びMinQtLog2SizeIntraYは、ＶＶＣドラフト６に定められているような様々な技法を使用して導出することができる。

【０１２８】

[0150] 代替的なやり方で、MinQtLog2SizeYの値は以下の等式２９を使用して導出することができる。

MinQtLog2SizeY=sps_log2_diff_min_qt_min_cb_luma+MinCbLog2SizeY （等式２９）

【０１２９】

[0151] sps_log2_diff_min_qt_min_cb_lumaは、図１８（例えば要素１８０１）に示すようにＳＰＳ内でシグナリングされ得る。MinCbLog2SizeYは、ＶＶＣドラフト６に定められているような様々な技法を使用して導出できることが理解されよう。

【０１３０】

[0152] SpsMaxMttDepthYに関して、或るやり方ではSpsMaxMttDepthYは
min(sps_max_mtt_hierarchy_depth_intra_slice_luma, sps_max_mtt_hierarchy_depth_inter_slice)
又は
max(sps_max_mtt_hierarchy_depth_intra_slice_luma, sps_max_mtt_hierarchy_depth_inter_slice)
として導出することができる。

【０１３１】

[0153] sps_max_mtt_hierarchy_depth_intra_slice_luma及びsps_max_mtt_hierarchy_depth_inter_sliceはＳＰＳ内でシグナリングされ得る。

【０１３２】

[0154] 代替的なやり方で、SpsMaxMttDepthYは
SpsMaxMttDepthY=sps_max_mtt_depth_luma （等式３０）
として導出することができる。

【０１３３】

[0155] sps_max_mtt_depth_lumaは、図１８（例えば要素１８０２）に示すようにＳＰＳ内でシグナリングされ得る。

【０１３４】

[0156] 更に、上記の例ではslice_cu_qp_delta_subdiv及びslice_cu_chroma_qp_offset_subdivの範囲がスライスヘッダの構文に依存する。例えばslice_cu_qp_delta_subdivの値の範囲は以下のように指定され得る。slice_typeがIに等しい場合、slice_cu_qp_delta_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeIntraY+SliceMaxMtt

DepthY)の範囲内にある。そうではない（slice_typeがIに等しくない）場合、slice_cu_qp_delta_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeInterY+SliceMaxMttDepthY)の範囲内にある。存在しない場合、slice_cu_qp_delta_subdivの値は０又はsps_cu_qp_delta_subdivに等しいと推論することができる。

【０１３５】

［0157］ slice_cu_chroma_qp_offset_subdivの値は以下のように指定され得る。slice_typeがIに等しい場合、slice_cu_chroma_qp_offset_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeIntraY+SliceMaxMttDepthY)の範囲内にある。そうではない（slice_typeがIに等しくない）場合、slice_cu_chroma_qp_offset_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeInterY+SliceMaxMttDepthY)の範囲内にある。存在しない場合、slice_cu_chroma_qp_offset_subdivの値は０又はsps_cu_chroma_qp_offset_subdivに等しいと推論することができる。

【０１３６】

［0158］ CtbLog2SizeY、MinQtLog2SizeIntraY、及びMinQtLog2SizeInterYが定められる場合、SliceMaxMttDepthYは

SliceMaxMttDepthY=slice_max_mtt_hierarchy_depth_luma （等式３１）

として導出することができる。

【０１３７】

［0159］ slice_max_mtt_hierarchy_depth_lumaはスライスヘッダ内でシグナリングされ得る。

【０１３８】

［0160］ 上記の例では、cu_qp_delta_subdivはslice_cu_qp_delta_subdivだと推論することができる。或いはcu_qp_delta_subdivはsps_cu_qp_delta_subdivだと最初に推論することができ、slice_cu_qp_delta_subdivが存在する場合、slice_cu_qp_delta_subdivがオーバライドし、cu_qp_delta_subdivはslice_cu_qp_delta_subdivだと推論することができる。Cu_qp_delta_subdivは$Qp_Y$を導出するために使用することができる。

【０１３９】

［0161］ 更に、上記の例ではcu_chroma_qp_offset_subdivをslice_cu_chroma_qp_offset_subdivだと推論することができる。或いはcu_chroma_qp_offset_subdivはsps_cu_chroma_qp_offset_subdivだと最初に推論することができ、slice_cu_chroma_qp_offset_subdivが存在する場合、slice_cu_chroma_qp_offset_subdivがオーバライドし、cu_chroma_qp_offset_subdivをslice_cu_chroma_qp_offset_subdivだと推論することができる。Cu_chroma_qp_offset_subdivは$Qp_{Cb}$、$Qp_{Cr}$、$QP_{CbCr}$を導出するために使用することができる。

【０１４０】

［0162］ 一部の実施形態では、cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivの構文がＰＰＳレベルにおいてシグナリングされ得る。しかし、cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivの範囲制限は、それらがスライス構文に依存しないように変更することができる。

【０１４１】

［0163］ 一例として、cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivは図５に示すようにＰＰＳ内でシグナリングされ得る。cu_qp_delta_subdivの値の範囲は以下のように指定することができる。cu_qp_delta_subdivの範囲は０～2*(CtbLog2SizeY-MinQtLog2SizeY+MaxMttDepthY)の範囲内にある。存在しない場合、cu_qp_delta_subdivの値は０に等しいと推論することができる。cu_chroma_qp_offset_subdivの値の範囲は以下のように指定することができる。cu_chroma_qp_offset_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeY+MaxMttDepthY)の範囲内にある。存在しない場合、cu_chroma_qp_offset_subdivの値は０に等しいと推論することができる。

【０１４２】

[0164] ctbLog2SizeYが定められる場合、MinQtLog2SizeY及びMaxMttDepthYはＳＰＳレベル上で推論することができる。例えばMaxMttDepthYは
min(sps_max_mtt_hierarchy_depth_intra_slice_luma, sps_max_mtt_hierarchy_depth_inter_slice)
又は
max(sps_max_mtt_hierarchy_depth_intra_slice_luma, sps_max_mtt_hierarchy_depth_inter_slice)
として導出することができる。

【０１４３】

[0165] sps_max_mtt_hierarchy_depth_intra_slice_luma及びsps_max_mtt_hierarchy_depth_inter_sliceはＳＰＳ内でシグナリングされ得る。

【０１４４】

[0166] 代替的なやり方で、MaxMttDepthYは
MaxMttDepthY=sps_max_mtt_depth_luma
として導出することができる。

【０１４５】

[0167] sps_max_mtt_depth_lumaは、図１９（例えば要素１９０１）に示すようにＳＰＳ内でシグナリングされ得る。

【０１４６】

[0168] 或るやり方では、MinQtLog2SizeYを
min(MinQtLog2SizeIntraY, MinQtLog2SizeInterY)
又は
max(MinQtLog2SizeIntraY, MinQtLog2SizeInterY)
として導出することができる。

【０１４７】

[0169] MinQtLog2SizeIntraY及びMinQtLog2SizeIntraYは、ＶＶＣドラフト６に定められているような様々な技法を使用して導出できることが理解されよう。

【０１４８】

[0170] 代替的なやり方で、MinQtLog2SizeYの値は以下の等式３２に基づいて導出することができる：
MinQtLog2SizeY=sps_log2_diff_min_qt_min_cb_lima+MinCbLog2SizeY （等式３２）

【０１４９】

[0171] sps_log2_diff_min_qt_min_cb_lumaは、図１３（例えば要素１３０１）に示すようにＳＰＳ内でシグナリングされる。MinCbLog2SizeYは、ＶＶＣドラフト６に定められているような様々な技法を使用して導出できることが理解されよう。

【０１５０】

[0172] 上記の例に基づき、cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivの範囲はスライスヘッダレベルにおいてオーバライドされない。

【０１５１】

[0173] 一部の実施形態では、cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivがＰＰＳレベルにおいてシグナリングされ得る。しかしcu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivの範囲制限は、それらがスライス構文に依存しないように固定することができる。

【０１５２】

[0174] 例えばcu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivは、図５に示すようにＰＰＳ内でシグナリングされ得る。cu_qp_delta_subdivの値の範囲は以下のように指定することができる。cu_qp_delta_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeY+MaxMttDepthY)の範囲内にある。存在しない場合、cu_qp_delta_subdivの値は０に等しいと推論することができる。cu_chroma_qp_offset_subdivの値の範

囲は以下のように指定することができる。cu_chroma_qp_offset_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeY+MaxMttDepthY)の範囲内にある。存在しない場合、cu_chroma_qp_offset_subdivの値は０に等しいと推論することができる。

【０１５３】

[0175] 但しCtbLog2SizeY、MinQtLog2SizeY、及びMaxMttDepthYは以下のやり方で導出することができる：CtbLog2SizeY/MinQtLog2SizeY/MaxMttDepthYはプロファイルによって指定することができ、又はCtbLog2SizeY/MinQtLog2SizeY/MaxMttDepthYは固定された数値であり得る。

【０１５４】

[0176] 上記の例に基づき、cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivの範囲はスライスヘッダレベルにおいてオーバライドされない。

【０１５５】

[0177] 一部の実施形態では、cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivは、図５に示すようにＰＰＳ内でシグナリングされ得る。cu_qp_delta_subdivの値の範囲は以下のように指定することができる。cu_qp_delta_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeY+MaxMttDepthY)の範囲内にある。存在しない場合、cu_qp_delta_subdivの値は０に等しいと推論することができる。cu_chroma_qp_offset_subdivの値の範囲は以下のように指定される。cu_chroma_qp_offset_subdivの値は０～2*(CtbLog2SizeY-MinQtLog2SizeY+MaxMttDepthY)の範囲内にある。存在しない場合、cu_chroma_qp_offset_subdivの値は０に等しいと推論することができる。

【０１５６】

[0178] CtbLog2SizeY、MinQtLog2SizeY、及びMaxMttDepthYはＰＰＳレベル上で推論することができる。例えば

CtbLog2SizeY=pps_log2_ctb_size （等式３３）

MinQtLog2SizeY=pps_log2_min_qt （等式３４）

MaxMttDepthY=pps_max_mtt_depth_luma （等式３５）

【０１５７】

[0179] pps_log2_ctb_size、pps_log2_min_qt、及びpps_max_mtt_depth_lumaは、図２０（例えば要素２００１）に示すようにＰＰＳ内でシグナリングされる。

【０１５８】

[0180] 上記の例に基づき、cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivの範囲はスライスヘッダレベルにおいてオーバライドされない。

【０１５９】

[0181] 図２１は、本開示の実施形態と合致する、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２１００の流れ図である。

【０１６０】

[0182] ステップ２１０２で、コード化ブロックの深度に関連する深度パラメータを受信することができる。深度パラメータは、例えばルマブロックのマルチタイプツリー階層の最大深度（例えば「slice_max_mtt_hierarchy_depth_luma」）から導出される変数「MaxMttDepthY」であり得る。一部の実施形態では、「slice_max_mtt_hierarchy_depth_luma」はコード化ブロックに関連するスライスヘッダ内でシグナリングされ得る。

【０１６１】

[0183] コード化ブロックはスライスに関連し得る。スライスはイントラ予測又はインター予測に関連することができる。スライスがイントラ予測に関連することに応じて、イントラ予測に関連するスライスについてデルタＱＰ値又はクロマＱＰオフセット値を決定することができる。そうではなくスライスがインター予測に関連することに応じて、インター予測に関連するスライスについてデルタＱＰ値又はクロマＱＰオフセット値を決定することができる。例えば「slice_type」が「I」に等しい（スライスがイントラ予測に関連することを示す）場合、「cu_qp_delta_subdiv」の値は０～2*(CtbLog2SizeY-MinQtLog2SizeIntraY+MaxMttDepthY)の範囲内にある。そうではなく「slice_type」が

「I」に等しくない（スライスがインター予測に関連することを示す）場合、「cu_qp_delta_subdiv」の値は０～2*(CtbLog2SizeY-MinQtLog2SizeInterY+MaxMttDepthY)の範囲内にある。更に一例として、「slice_type」が「I」に等しい場合、「cu_chroma_qp_offset_subdiv」の値は０～2*(CtbLog2SizeY-MinQtLog2SizeIntraY+MaxMttDepthY)の範囲内にある。そうではなく「slice_type」が「I」に等しくない場合、「cu_chroma_qp_offset_subdiv」の値は０～2*(CtbLog2SizeY-MinQtLog2SizeInterY+MaxMttDepthY)の範囲内にある。

【０１６２】

［0184］一部の実施形態では、深度パラメータがピクチャヘッダ内でシグナリングされ得る。ピクチャは複数のスライスを含み得ることが理解されよう。イントラ予測に関連するスライスでは、イントラ予測に関連するスライスについて対応するデルタＱＰ値又はクロマＱＰオフセット値を決定することができる。インター予測に関連するスライスでは、インター予測に関連するスライスについて対応するデルタＱＰ値又はクロマＱＰオフセット値を決定することができる。例えば図１１の表１１に関して論じるように、イントラ予測に関連するスライスについてデルタＱＰ値及びクロマＱＰオフセット値を導出するために、ph_cu_qp_delta_subdiv_intra_slice及びph_cu_chroma_qp_offset_subdiv_intra_sliceがピクチャヘッダ内でシグナリングされる。インター予測に関連するスライスについてデルタＱＰ値及びクロマＱＰオフセット値を導出するために、ph_cu_qp_delta_subdiv_inter_slice及びph_cu_chroma_qp_offset_subdiv_inter_sliceがピクチャヘッダ内でシグナリングされる。

【０１６３】

［0185］ステップ２１０４で、コード化ブロックの深度に基づいてデルタ量子化パラメータ（ＱＰ）値又はクロマＱＰオフセット値の少なくとも１つを決定することができる。上記で論じたように、デルタＱＰ値は「cu_qp_delta_subdiv」に基づいて決定することができ、クロマＱＰオフセット値は「cu_chroma_qp_offset_subdiv」に基づいて決定することができ、「cu_qp_delta_subdiv」及び「cu_chroma_qp_offset_subdiv」は変数「MaxMttDepthY」に基づいて決定することができる。

【０１６４】

［0186］ステップ２１０６で、決定したデルタＱＰ値に基づいてルマＱＰ値を導出することができ、決定したクロマＱＰオフセット値に基づいてクロマＱＰ値を導出することができる。

【０１６５】

［0187］ステップ２１０８で、導出したルマＱＰ値及び導出したクロマＱＰ値に基づいてコード化ブロックを処理することができる。

【０１６６】

［0188］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の装置（開示する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（登録商標）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及び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。

【０１６７】

［0189］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．コンピュータによって実装される方法であって、

コード化映像データを含むビットストリームを受信すること、

コード化ブロックの第１のパラメータを決定すること、

デルタ量子化パラメータ（ＱＰ）値又はクロマＱＰオフセット値に関連する１つ又は複数の第２のパラメータを第１のパラメータに従って決定すること、及び
デルタＱＰ値又はクロマＱＰオフセット値の少なくとも１つを１つ又は複数の第２のパラメータに従って決定すること
を含む、方法。
２．コード化ブロックの第１のパラメータを決定することは、
コード化ブロックがイントラ予測スライス又はインター予測スライスに関連するかどうかを判定すること、及び
コード化ブロックがイントラ予測スライスに関連することに応じて、第１のパラメータをイントラ予測スライスに関連するパラメータであるように決定すること、又は
コード化ブロックがインター予測スライスに関連することに応じて、第１のパラメータをインター予測スライスに関連するパラメータであるように決定すること
を含む、条項１に記載の方法。
３．第１のパラメータがコード化ブロックに関連するスライスヘッダ内でシグナリングされる、条項１に記載の方法。
４．第１のパラメータがコード化ブロックに関連するピクチャヘッダ内でシグナリングされる、条項１に記載の方法。
５．デルタＱＰ値に基づいてルマＱＰ値を決定すること、
クロマＱＰオフセット値に基づいてクロマＱＰ値を決定すること、及び
ルマＱＰ値及びクロマＱＰ値に基づいてコード化ブロックを処理すること
を更に含む、条項１に記載の方法。
６．映像コンテンツを処理するためのシステムであって、
１組の命令を記憶するメモリと、
少なくとも１つのプロセッサとを含み、少なくとも１つのプロセッサは、
コード化映像データを含むビットストリームを受信すること、
コード化ブロックの第１のパラメータを決定すること、
デルタ量子化パラメータ（ＱＰ）値又はクロマＱＰオフセット値に関連する１つ又は複数の第２のパラメータを第１のパラメータに従って決定すること、及び
デルタＱＰ値又はクロマＱＰオフセット値の少なくとも１つを１つ又は複数の第２のパラメータに従って決定すること
をシステムに行わせるように、１組の命令を実行するように構成される、システム。
７．少なくとも１つのプロセッサが、
コード化ブロックがイントラ予測スライス又はインター予測スライスに関連するかどうかを判定すること、及び
コード化ブロックがイントラ予測スライスに関連することに応じて、第１のパラメータをイントラ予測スライスに関連するパラメータであるように決定すること、又は
コード化ブロックがインター予測スライスに関連することに応じて、第１のパラメータをインター予測スライスに関連するパラメータであるように決定すること
をシステムに更に行わせるように、１組の命令を実行するように構成される、条項６に記載のシステム。
８．第１のパラメータがコード化ブロックに関連するスライスヘッダ内でシグナリングされる、条項６に記載のシステム。
９．第１のパラメータがコード化ブロックに関連するピクチャヘッダ内でシグナリングされる、条項６に記載のシステム。
１０．少なくとも１つのプロセッサが、
デルタＱＰ値に基づいてルマＱＰ値を決定すること、
クロマＱＰオフセット値に基づいてクロマＱＰ値を決定すること、及び
ルマＱＰ値及びクロマＱＰ値に基づいてコード化ブロックを処理すること
をシステムに更に行わせるように、１組の命令を実行するように構成される、条項６に記載のシステム。

１１．コンピュータシステムの少なくとも１つのプロセッサによって実行可能な命令を記憶する非一時的コンピュータ可読媒体であって、命令を実行することが、
コード化映像データを含むビットストリームを受信すること、
コード化ブロックの第１のパラメータを決定すること、
デルタ量子化パラメータ（ＱＰ）値又はクロマＱＰオフセット値に関連する１つ又は複数の第２のパラメータを第１のパラメータに従って決定すること、及び
デルタＱＰ値又はクロマＱＰオフセット値の少なくとも１つを１つ又は複数の第２のパラメータに従って決定すること
を含む方法をコンピュータシステムに行わせる、非一時的コンピュータ可読媒体。
１２．方法は、
コード化ブロックがイントラ予測スライス又はインター予測スライスに関連するかどうかを判定すること、及び
コード化ブロックがイントラ予測スライスに関連することに応じて、第１のパラメータをイントラ予測スライスに関連するパラメータであるように決定すること、又は
コード化ブロックがインター予測スライスに関連することに応じて、第１のパラメータをインター予測スライスに関連するパラメータであるように決定すること
を更に含む、条項１１に記載の非一時的コンピュータ可読媒体。
１３．第１のパラメータがコード化ブロックに関連するスライスヘッダ内でシグナリングされる、条項１１に記載の非一時的コンピュータ可読媒体。
１４．第１のパラメータがコード化ブロックに関連するピクチャヘッダ内でシグナリングされる、条項１１に記載の非一時的コンピュータ可読媒体。
１５．方法が、
デルタＱＰ値に基づいてルマＱＰ値を決定すること、
クロマＱＰオフセット値に基づいてクロマＱＰ値を決定すること、及び
ルマＱＰ値及びクロマＱＰ値に基づいてコード化ブロックを処理すること
を更に含む、条項１１に記載の非一時的コンピュータ可読媒体。

【０１６８】

[0190] 本明細書の「第１の」及び「第２の」等の関係語は、あ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されるに過ぎず、それらのエンティティ又は操作間のいかなる実際の関係又は順序も必要としないか又は含意しないことに留意すべきである。更に、「含む」、「有する」、「含有する」及び「包含する」並びに他の同様の形式の用語は、意味の点で均等であることを意図し、これらの用語の何れか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いないか、又は列挙するアイテムのみ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。

【０１６９】

[0191] 本明細書で使用するとき、別段の定めがない限り、「又は」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、あるデータベースがＡ又はＢを含み得ると述べた場合、別段の定めがない限り又は実行不可能でない限り、そのデータベースは、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、あるデータベースがＡ、Ｂ又はＣを含み得ると述べた場合、別段の定めがない限り又は実行不可能でない限り、そのデータベースは、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。

【０１７０】

[0192] 上記で説明した実施形態は、ハードウェア若しくはソフトウェア（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るであろう。ソフトウェアによって実装される場合、ソフトウェアは、上記のコンピュータ可読媒体に記憶することができる。ソフトウェアは、プロセッサによって実行されるとき、開示する方法を実行することができる。本開示で説明した計算ユニット及び他の機能ユニットは、ハードウェア若しく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／ユニットの複数を１つのモジュール／ユニットとして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／サブユニットに更に分割できることも当業者であれば理解するであろう。

【０１７１】

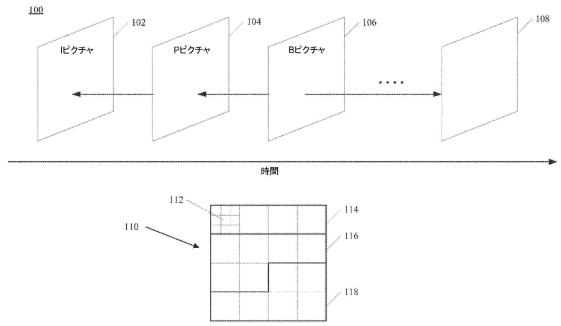
［0193］上記の本明細書では、実装形態ごとに変わり得る多数の具体的な詳細に関して実施形態を説明してきた。記載した実施形態に対する一定の適応形態及び修正形態がなされ得る。本明細書を検討し、本明細書で開示する本発明を実践することで他の実施形態が当業者に明らかになり得る。本明細書及び例は、専ら例示として検討され、本開示の真の範囲及び趣旨は、添付の特許請求の範囲によって示されることを意図する。図中に示すステップの順序は、例示目的に過ぎず、特定のステップの順序に限定されることを意図しない。そのため、それらのステップは、同じ方法を実装しながら異なる順序で実行できることを当業者であれば理解することができる。

【０１７２】

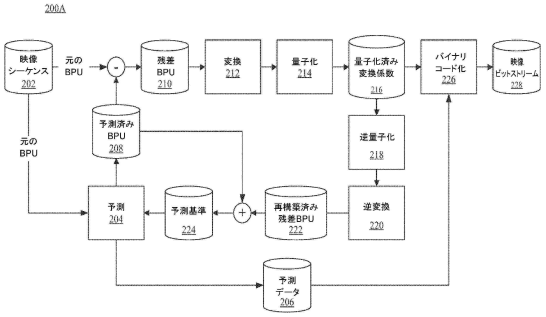
［0194］図面及び本明細書で例示的実施形態を開示してきた。しかし、それらの実施形態に対する多くの改変形態及び修正形態がなされ得る。従って、特定の用語を使用したが、それらの用語は、限定目的ではなく、全般的及び説明的な意味で使用されたものに過ぎない。

【図面】

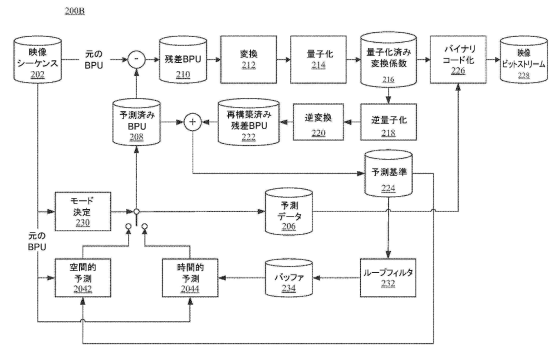
【図１】



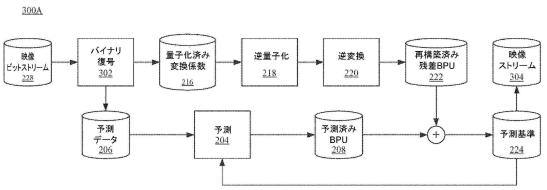

【図２Ａ】



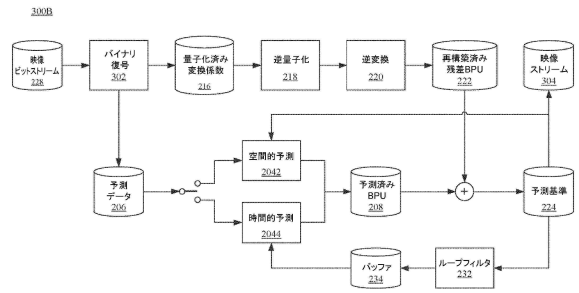

【図２Ｂ】



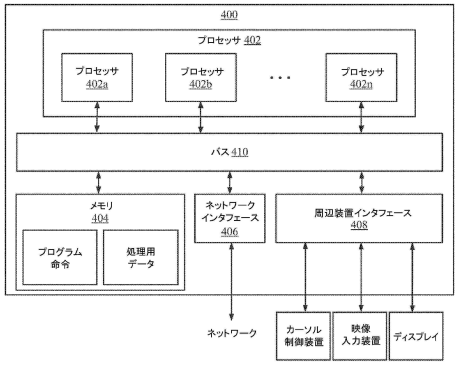

【図３Ａ】



【図３Ｂ】




【図４】




【図５】

表5:CUデルタQPのためのPPS構文

| pic_parameter_set_rbsp( ) { | 記述子 |
|---|---|
| ... | |
| | |
| **cu_qp_delta_enabled_flag** | u(1) |
| if( cu_qp_delta_enabled_flag ) | |
| **cu_qp_delta_subdiv** | ue(v) |
| ... | |
| **cu_chroma_qp_offset_enabled_flag** | u(1) |
| if( cu_chroma_qp_offset_enabled_flag ) { | |
| **cu_chroma_qp_offset_subdiv** | ue(v) |
| .... | |

【図６Ａ】

表6: CUデルタQPのためのコード化ツリーレベル構文

| coding_tree( x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mttDepth, depthOffset, partIdx, treeTypeCurr, modeTypeCurr ) { | 記述子 |
|---|---|
| if( ( allowSplitBtVer \|\| allowSplitBtHor \|\| allowSplitTtVer \|\| allowSplitTtHor \|\| allowSplitQt ) && ( x0 + cbWidth <= pps_pic_width_in_luma_samples ) && ( y0 + cbHeight <= pps_pic_height_in_luma_samples ) ) | |
| **split_cu_flag** | ae(v) |
| if( pps_cu_qp_delta_enabled_flag && qgOnY && cbSubdiv <= CuQpDeltaSubdiv ) { | |
| IsCuQpDeltaCoded = 0 | |
| CuQpDeltaVal = 0 | |
| CuQgTopLeftX = x0 | |
| CuQgTopLeftY = y0 | |
| } | |
| if( sh_cu_chroma_qp_offset_enabled_flag && qgOnC && cbSubdiv <= CuChromaQpOffsetSubdiv ) { | |
| IsCuChromaQpOffsetCoded = 0 | |
| $CuQpOffset_{Cb}$ = 0 | |
| $CuQpOffset_{Cr}$ = 0 | |
| $CuQpOffset_{CbCr}$ = 0 | |
| } | |
| if( split_cu_flag ) { | |
| if( ( allowSplitBtVer \|\| allowSplitBtHor \|\| allowSplitTtVer \|\| allowSplitTtHor ) && allowSplitQt ) | |
| **split_qt_flag** | ae(v) |
| if( !split_qt_flag ) { | |
| if( ( allowSplitBtHor \|\| allowSplitTtHor ) && ( allowSplitBtVer \|\| allowSplitTtVer ) ) | |
| **mtt_split_cu_vertical_flag** | ae(v) |
| if( ( allowSplitBtVer && allowSplitTtVer && mtt_split_cu_vertical_flag ) \|\| ( allowSplitBtHor && allowSplitTtHor && !mtt_split_cu_vertical_flag ) ) | |
| **mtt_split_cu_binary_flag** | ae(v) |
| } | |
| if( ModeTypeCondition == 1 ) | |
| modeType = MODE_TYPE_INTRA | |
| else if( ModeTypeCondition == 2 ) { | |
| non_inter_flag | ae(v) |
| modeType = non_inter_flag ? MODE_TYPE_INTRA : MODE_TYPE_INTER | |
| } else | |
| modeType = modeTypeCurr | |
| treeType = ( modeType == MODE_TYPE_INTRA ) ? DUAL_TREE_LUMA : treeTypeCurr | |

【図６Ｂ】

表6: CUデルタQPのためのコード化ツリーレベル構文(続き)

| | |
|---|---|
| if( !split_qt_flag ) { | |
| if( MttSplitMode[ x0 ][ y0 ][ mttDepth ] == SPLIT_BT_VER ) { | |
| depthOffset += ( x0 + cbWidth > pps_pic_width_in_luma_samples ) ? 1 : 0 | |
| x1 = x0 + ( cbWidth / 2 ) | |
| coding_tree( x0, y0, cbWidth / 2, cbHeight, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType ) | |
| if( x1 < pps_pic_width_in_luma_samples ) | |
| coding_tree( x1, y0, cbWidth / 2, cbHeight, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType ) | |
| } else if( MttSplitMode[ x0 ][ y0 ][ mttDepth ] == SPLIT_BT_HOR ) { | |
| depthOffset += ( y0 + cbHeight > pps_pic_height_in_luma_samples ) ? 1 : 0 | |
| y1 = y0 + ( cbHeight / 2 ) | |
| coding_tree( x0, y0, cbWidth, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType ) | |
| if( y1 < pps_pic_height_in_luma_samples ) | |
| coding_tree( x0, y1, cbWidth, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType ) | |
| } else if( MttSplitMode[ x0 ][ y0 ][ mttDepth ] == SPLIT_TT_VER ) { | |
| x1 = x0 + ( cbWidth / 4 ) | |
| x2 = x0 + ( 3 * cbWidth / 4 ) | |
| qgNextOnY = qgOnY && ( cbSubdiv + 2 <= CuQpDeltaSubdiv ) | |
| qgNextOnC = qgOnC && ( cbSubdiv + 2 <= CuChromaQpOffsetSubdiv ) | |
| coding_tree( x0, y0, cbWidth / 4, cbHeight, qgNextOnY, qgNextOnC, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType ) | |
| coding_tree( x1, y0, cbWidth / 2, cbHeight, qgNextOnY, qgNextOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType ) | |
| coding_tree( x2, y0, cbWidth / 4, cbHeight, qgNextOnY, qgNextOnC, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType ) | |
| } else { /* SPLIT_TT_HOR */ | |
| y1 = y0 + ( cbHeight / 4 ) | |
| y2 = y0 + ( 3 * cbHeight / 4 ) | |
| qgNextOnY = qgOnY && ( cbSubdiv + 2 <= CuQpDeltaSubdiv ) | |
| qgNextOnC = qgOnC && ( cbSubdiv + 2 <= CuChromaQpOffsetSubdiv ) | |
| coding_tree( x0, y0, cbWidth, cbHeight / 4, qgNextOnY, qgNextOnC, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 0, treeType, modeType ) | |
| coding_tree( x0, y1, cbWidth, cbHeight / 2, qgNextOnY, qgNextOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType ) | |

【図６Ｃ】

表6: CUデルタQPのためのコード化ツリーレベル構文(続き)

| | |
|---|---|
| coding_tree( x0, y2, cbWidth, cbHeight / 4, qgNextOnY, qgNextOnC, cbSubdiv + 2, cqtDepth, mttDepth + 1, depthOffset, 2, treeType, modeType ) | |
| } | |
| } else { | |
| x1 = x0 + ( cbWidth / 2 ) | |
| y1 = y0 + ( cbHeight / 2 ) | |
| coding_tree( x0, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 0, treeType, modeType ) | |
| if( x1 < pps_pic_width_in_luma_samples ) | |
| coding_tree( x1, y0, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 1, treeType, modeType ) | |
| if( y1 < pps_pic_height_in_luma_samples ) | |
| coding_tree( x0, y1, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 2, treeType, modeType ) | |
| if( y1 < pps_pic_height_in_luma_samples && x1 < pps_pic_width_in_luma_samples ) | |
| coding_tree( x1, y1, cbWidth / 2, cbHeight / 2, qgOnY, qgOnC, cbSubdiv + 2, cqtDepth + 1, 0, 0, 3, treeType, modeType ) | |
| } | |
| if( modeTypeCurr == MODE_TYPE_ALL && modeType == MODE_TYPE_INTRA ) | |
| coding_tree( x0, y0, cbWidth, cbHeight, 0, qgOnC, cbSubdiv, cqtDepth, mttDepth, 0, 0, DUAL_TREE_CHROMA , modeType ) | |
| } else | |
| coding_unit( x0, y0, cbWidth, cbHeight, cqtDepth, treeTypeCurr , modeTypeCurr ) | |
| } | |

10

20

【図７Ａ】

表7: CUデルタQPのための変換単位レベル構文

| | 記述子 |
|---|---|
| transform_unit( x0, y0, tbWidth, tbHeight, treeType, subTuIndex, cbType ) { | |
| if( IntraSubPartitionsSplitType != ISP_NO_SPLIT && subTuIndex == NumIntraSubPartitions - 1 ) { | |
| xC = CbPosX[ cbType ][ x0 ][ y0 ] | |
| yC = CbPosY[ cbType ][ x0 ][ y0 ] | |
| wC = CbWidth[ cbType ][ x0 ][ y0 ] >> SubWidthC | |
| hC = CbHeight[ cbType ][ x0 ][ y0 ] >> SubHeightC | |
| } else { | |
| xC = x0 | |
| yC = y0 | |
| wC = tbWidth >> SubWidthC | |
| hC = tbHeight >> SubHeightC | |
| } | |
| chromaAvailable = treeType != DUAL_TREE_LUMA && sps_chroma_format_idc != 0 && ( IntraSubPartitionsSplitType == ISP_NO_SPLIT || ( subTuIndex == NumIntraSubPartitions - 1 ) ) | |
| if( treeType == SINGLE_TREE \|\| treeType == DUAL_TREE_CHROMA ) && sps_chroma_format_idc != 0 && ( IntraSubPartitionsSplitType == ISP_NO_SPLIT && !( cu_sbt_flag && ( subTuIndex == 0 && cu_sbt_pos_flag ) \|\| ( subTuIndex == 1 && !cu_sbt_pos_flag ) ) ) \|\| ( IntraSubPartitionsSplitType != ISP_NO_SPLIT && ( subTuIndex == NumIntraSubPartitions - 1 ) ) ) { | |
| tu_cb_coded_flag[ xC ][ yC ] | ae(v) |
| tu_cr_coded_flag[ xC ][ yC ] | ae(v) |
| } | |
| if( treeType == SINGLE_TREE \|\| treeType == DUAL_TREE_LUMA ) { | |
| if( ( IntraSubPartitionsSplitType == ISP_NO_SPLIT && !( cu_sbt_flag && ( ( subTuIndex == 0 && cu_sbt_pos_flag ) \|\| ( subTuIndex == 1 && !cu_sbt_pos_flag ) ) && ( sps_sbt_max_size_64_flag && tbWidth <= 64 && tbHeight <= 64 \|\| !sps_sbt_max_size_64_flag ) ) ) \|\| ( CuPredMode[ cbType ][ x0 ][ y0 ] == MODE_INTRA && sps_sbt_enabled_flag && !ciip_flag[ x0 ][ y0 ] && subTuIndex == NumIntraSubPartitions - 1 ) ) { | |
| tu_y_coded_flag[ xC ][ yC ] = MaxTbSizeY \|\| | |
| CbWidth[ cbType ][ x0 ][ y0 ] > MaxTbSizeY \|\| | |
| CbHeight[ cbType ][ x0 ][ y0 ] > MaxTbSizeY ) { | |
| ( IntraSubPartitionsSplitType != ISP_NO_SPLIT && ( subTuIndex < NumIntraSubPartitions - 1 \|\| !InferTuCbf[ xC ][ yC ] ) ) | |
| tu_y_coded_flag[ x0 ][ y0 ] | ae(v) |
| if( IntraSubPartitionsSplitType != ISP_NO_SPLIT ) | |
| InferTuCbf[ xC ][ yC ] = tu_y_coded_flag[ x0 ][ y0 ] | |
| } | |

【図７Ｂ】

表7: CUデルタQPのための変換単位レベル構文

| | |
|---|---|
| if( ( CbWidth[ chType ][ x0 ][ y0 ] > 64 \|\| CbHeight[ chType ][ x0 ][ y0 ] > 64 \|\| tu_y_coded_flag[ x0 ][ y0 ] \|\| ( chromaAvailable && ( tu_cb_coded_flag[ xC ][ yC ] \|\| tu_cr_coded_flag[ xC ][ yC ] ) ) && treeType != DUAL_TREE_CHROMA && pps_cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded ) { | |
| **cu_qp_delta_abs** | ae(v) |
| if( cu_qp_delta_abs ) | |
| **cu_qp_delta_sign_flag** | ae(v) |
| } | |
| if( ( CbWidth[ chType ][ x0 ][ y0 ] > 64 \|\| CbHeight[ chType ][ x0 ][ y0 ] > 64 \|\| ( chromaAvailable && ( tu_cb_coded_flag[ xC ][ yC ] \|\| tu_cr_coded_flag[ xC ][ yC ] ) ) ) && treeType != DUAL_TREE_LUMA && sh_cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded ) { | |
| **cu_chroma_qp_offset_flag** | ae(v) |
| if( cu_chroma_qp_offset_flag && pps_chroma_qp_offset_list_len_minus1 > 0 ) | |
| **cu_chroma_qp_offset_idx** | ae(v) |
| } | |
| if( sps_joint_cbcr_enabled_flag && ( ( CuPredMode[ chType ][ x0 ][ y0 ] == MODE_INTRA && ( tu_cb_coded_flag[ xC ][ yC ] \|\| tu_cr_coded_flag[ xC ][ yC ] ) ) \|\| ( tu_cb_coded_flag[ xC ][ yC ] && tu_cr_coded_flag[ xC ][ yC ] ) ) && chromaAvailable ) | |
| **tu_joint_cbcr_residual_flag**[ xC ][ yC ] | ae(v) |
| if( tu_y_coded_flag[ x0 ][ y0 ] && treeType != DUAL_TREE_CHROMA ) { | |
| if( sps_transform_skip_enabled_flag && !BdpcmFlag[ x0 ][ y0 ][ 0 ] && tbWidth <= MaxTsSize && tbHeight <= MaxTsSize && ( IntraSubPartitionsSplitType == ISP_NO_SPLIT ) && !cu_sbt_flag ) | |
| **transform_skip_flag**[ x0 ][ y0 ][ 0 ] | ae(v) |
| if( !transform_skip_flag[ x0 ][ y0 ][ 0 ] \|\| sh_ts_residual_coding_disabled_flag ) | |
| residual_coding( x0, y0, Log2( tbWidth ), Log2( tbHeight ), 0 ) | |
| else | |
| residual_ts_coding( x0, y0, Log2( tbWidth ), Log2( tbHeight ), 0 ) | |
| } | |
| if( tu_cb_coded_flag[ xC ][ yC ] && treeType != DUAL_TREE_LUMA ) { | |
| if( sps_transform_skip_enabled_flag && !BdpcmFlag[ x0 ][ y0 ][ 1 ] && wC <= MaxTsSize && hC <= MaxTsSize && !cu_sbt_flag ) | |
| **transform_skip_flag**[ xC ][ yC ][ 1 ] | ae(v) |
| if( !transform_skip_flag[ xC ][ yC ][ 1 ] \|\| sh_ts_residual_coding_disabled_flag ) | |
| residual_coding( xC, yC, Log2( wC ), Log2( hC ), 1 ) | |
| else | |
| residual_ts_coding( xC, yC, Log2( wC ), Log2( hC ), 1 ) | |
| } | |

【図７Ｃ】

表7: CUデルタQPのための変換単位レベル構文(続き)

| | |
|---|---|
| if( tu_cb_coded_flag[ xC ][ yC ] && treeType != DUAL_TREE_LUMA ) { | |
| if( sps_transform_skip_enabled_flag && !BdpcmFlag[ x0 ][ y0 ][ 1 ] && wC <= MaxTsSize && hC <= MaxTsSize && !cu_sbt_flag ) | |
| **transform_skip_flag**[ xC ][ yC ][ 1 ] | ae(v) |
| if( !transform_skip_flag[ xC ][ yC ][ 1 ] \|\| sh_ts_residual_coding_disabled_flag ) | |
| residual_coding( xC, yC, Log2( wC ), Log2( hC ), 1 ) | |
| else | |
| residual_ts_coding( xC, yC, Log2( wC ), Log2( hC ), 1 ) | |
| } | |
| if( tu_cr_coded_flag[ xC ][ yC ] && treeType != DUAL_TREE_LUMA && !( tu_cb_coded_flag[ xC ][ yC ] && tu_joint_cbcr_residual_flag[ xC ][ yC ] ) ) { | |
| if( sps_transform_skip_enabled_flag && !BdpcmFlag[ x0 ][ y0 ][ 2 ] && wC <= MaxTsSize && hC <= MaxTsSize && !cu_sbt_flag ) | |
| **transform_skip_flag**[ xC ][ yC ][ 2 ] | ae(v) |
| if( !transform_skip_flag[ xC ][ yC ][ 2 ] \|\| sh_ts_residual_coding_disabled_flag ) | |
| residual_coding( xC, yC, Log2( wC ), Log2( hC ), 2 ) | |
| else | |
| residual_ts_coding( xC, yC, Log2( wC ), Log2( hC ), 2 ) | |
| } | |
| } | |

【図８】

表8: スライスヘッダ構文

| slice_header( ) { | **記述子** |
|---|---|
| … | |
| if( partition_constraints_override_enabled_flag ) { | |
| **partition_constraints_override_flag** | ue(v) |
| if( partition_constraints_override_flag ) { | |
| **slice_log2_diff_min_qt_min_cb_luma** | ue(v) |
| **slice_max_mtt_hierarchy_depth_luma** | ue(v) |
| if( slice_max_mtt_hierarchy_depth_luma != 0 ) | |
| **slice_log2_diff_max_bt_min_qt_luma** | ue(v) |
| **slice_log2_diff_max_tt_min_qt_luma** | ue(v) |
| } | |

【図９】

表9: スライスヘッダ構文

| slice_header( ) { | **記述子** |
|---|---|
| .. | |
| if( partition_constraints_override_enabled_flag ) { | |
| **partition_constraints_override_flag** | ue(v) |
| if( partition_constraints_override_flag ) { | |
| **slice_log2_diff_min_qt_min_cb_luma** | ue(v) |
| **slice_max_mtt_hierarchy_depth_luma** | ue(v) |
| if( slice_max_mtt_hierarchy_depth_luma != 0 ){ | |
| **slice_log2_diff_max_bt_min_qt_luma** | ue(v) |
| **slice_log2_diff_max_tt_min_qt_luma** | ue(v) |
| } | |
| if( slice_type = = I && qtbtt_dual_tree_intra_flag ) { | |
| **slice_log2_diff_min_qt_min_cb_chroma** | ue(v) |
| **slice_max_mtt_hierarchy_depth_chroma** | ue(v) |
| if( slice_max_mtt_hierarchy_depth_chroma != 0 ){ | |
| **slice_log2_diff_max_bt_min_qt_chroma** | ue(v) |
| **slice_log2_diff_max_tt_min_qt_chroma** | ue(v) |
| } | |
| } | |
| } | |
| } | |
| *if( cu_qp_delta_enabled_flag )* | |
| ***cu_qp_delta_subdiv*** | ***ue(v)*** |
| *if( cu_chroma_qp_offset_enabled_flag ) {* | |
| ***cu_chroma_qp_offset_subdiv*** | ***ue(v)*** |
| | |

901

【図１０】

表10: スライスヘッダ構文

| slice_header( ) { | **記述子** |
|---|---|
| .. | |
| if( partition_constraints_override_enabled_flag ) { | |
| **partition_constraints_override_flag** | ue(v) |
| if( partition_constraints_override_flag ) { | |
| **slice_log2_diff_min_qt_min_cb_luma** | ue(v) |
| **slice_max_mtt_hierarchy_depth_luma** | ue(v) |
| if( slice_max_mtt_hierarchy_depth_luma != 0 ){ | |
| **slice_log2_diff_max_bt_min_qt_luma** | ue(v) |
| **slice_log2_diff_max_tt_min_qt_luma** | ue(v) |
| } | |
| if( slice_type = = I && qtbtt_dual_tree_intra_flag ) { | |
| **slice_log2_diff_min_qt_min_cb_chroma** | ue(v) |
| **slice_max_mtt_hierarchy_depth_chroma** | ue(v) |
| if( slice_max_mtt_hierarchy_depth_chroma != 0 ){ | |
| **slice_log2_diff_max_bt_min_qt_chroma** | ue(v) |
| **slice_log2_diff_max_tt_min_qt_chroma** | ue(v) |
| } | |
| } | |
| } | |
| } | |
| ***cu_qp_delta_enabled_flag*** | ***u(1)*** |
| *if( cu_qp_delta_enabled_flag )* | |
| ***cu_qp_delta_subdiv*** | ***ue(v)*** |
| ***cu_chroma_qp_offset_enabled_flag*** | ***u(1)*** |
| *if( cu_chroma_qp_offset_enabled_flag ) {* | |
| ***cu_chroma_qp_offset_subdiv*** | ***ue(v)*** |
| | |

1001

【図１１】

表11: 例示的なピクチャヘッダ構文

| picture_header_structure( ) { | 記述子 | |
|---|---|---|
| … | u(1) | |
| **ph_inter_slice_allowed_flag** | u(1) | |
| if( ph_inter_slice_allowed_flag ) | | |
| **ph_intra_slice_allowed_flag** | u(1) | |
| … | | |
| if( sps_partition_constraints_override_enabled_flag ) | | |
| **ph_partition_constraints_override_flag** | u(1) | |
| if( ph_intra_slice_allowed_flag ) { | | |
| if( ph_partition_constraints_override_flag ) { | | |
| **ph_log2_diff_min_qt_min_cb_intra_slice_luma** | ue(v) | |
| **ph_max_mtt_hierarchy_depth_intra_slice_luma** | ue(v) | |
| if( ph_max_mtt_hierarchy_depth_intra_slice_luma != 0 ) { | | |
| **ph_log2_diff_max_bt_min_qt_intra_slice_luma** | ue(v) | |
| **ph_log2_diff_max_tt_min_qt_intra_slice_luma}** | ue(v) | |
| if( sps_qtbtt_dual_tree_intra_flag ) { | | |
| **ph_log2_diff_min_qt_min_cb_intra_slice_chroma** | ue(v) | |
| **ph_max_mtt_hierarchy_depth_intra_slice_chroma** | ue(v) | |
| if( ph_max_mtt_hierarchy_depth_intra_slice_chroma != 0 ) { | | |
| **ph_log2_diff_max_bt_min_qt_intra_slice_chroma** | ue(v) | |
| **ph_log2_diff_max_tt_min_qt_intra_slice_chroma** | ue(v) | |
| }}} | | |
| if( pps_cu_qp_delta_enabled_flag ) | | |
| *ph_cu_qp_delta_subdiv_intra_slice* | ue(v) | }1101 |
| if( pps_cu_chroma_qp_offset_list_enabled_flag ) | | |
| *ph_cu_chroma_qp_offset_subdiv_intra_slice* | ue(v) | }1102 |
| } | | |
| if( ph_inter_slice_allowed_flag ) { | | |
| if( ph_partition_constraints_override_flag ) { | | |
| **ph_log2_diff_min_qt_min_cb_inter_slice** | ue(v) | |
| **ph_max_mtt_hierarchy_depth_inter_slice** | ue(v) | |
| if( ph_max_mtt_hierarchy_depth_inter_slice != 0 ) { | | |
| **ph_log2_diff_max_bt_min_qt_inter_slice** | ue(v) | |
| **ph_log2_diff_max_tt_min_qt_inter_slice** }} | ue(v) | |
| if( pps_cu_qp_delta_enabled_flag ) | | |
| *ph_cu_qp_delta_subdiv_inter_slice* | ue(v) | }1103 |
| if( pps_cu_chroma_qp_offset_list_enabled_flag ) | | |
| *ph_cu_chroma_qp_offset_subdiv_inter_slice* | ue(v) | }1104 |
| …} | | |
| if( pps_qp_delta_info_in_ph_flag ) | | |
| **ph_qp_delta** | se(v) | |
| …} | | |

【図１２】

表12: cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivのためのPPS構文

| pic_parameter_set_rbsp( ) { | 記述子 | |
|---|---|---|
| … | | |
| | | |
| **cu_qp_delta_enabled_flag** | u(1) | |
| if( cu_qp_delta_enabled_flag ) | | |
| *pps_cu_qp_delta_subdiv* | ue(v) | }1201 |
| … | | |
| **cu_chroma_qp_offset_enabled_flag** | u(1) | |
| if( cu_chroma_qp_offset_enabled_flag ) { | | |
| *pps_cu_chroma_qp_offset_subdiv* | ue(v) | }1202 |
| … | | |

【図１３】

表13: cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivのためのスライスヘッダ構文

| slice_header( ) { | 記述子 | |
|---|---|---|
| .. | | |
| if( partition_constraints_override_enabled_flag ) { | | |
| **partition_constraints_override_flag** | ue(v) | |
| if( partition_constraints_override_flag ) { | | |
| **slice_log2_diff_min_qt_min_cb_luma** | ue(v) | |
| **slice_max_mtt_hierarchy_depth_luma** | ue(v) | |
| if( slice_max_mtt_hierarchy_depth_luma != 0 ){ | | |
| **slice_log2_diff_max_bt_min_qt_luma** | ue(v) | |
| **slice_log2_diff_max_tt_min_qt_luma** | ue(v) | |
| } | | |
| if( slice_type = = I && qtbtt_dual_tree_intra_flag ) { | | |
| **slice_log2_diff_min_qt_min_cb_chroma** | ue(v) | |
| **slice_max_mtt_hierarchy_depth_chroma** | ue(v) | |
| if( slice_max_mtt_hierarchy_depth_chroma != 0 ){ | | |
| **slice_log2_diff_max_bt_min_qt_chroma** | ue(v) | |
| **slice_log2_diff_max_tt_min_qt_chroma** | ue(v) | |
| } | | |
| } | | |
| } | | |
| } | | |
| *if( cu_qp_delta_enabled_flag && partition_constraints_override_flag)* | | |
| *slice_cu_qp_delta_subdiv* | *ue(v)* | }1301 |
| *if( cu_chroma_qp_offset_enabled_flag && partition_constraints_override_flag) {* | | |
| *slice_cu_chroma_qp_offset_subdiv* | *ue(v)* | |
| | | |

【図１４】

表14: sps_max_mtt_depth_lumaのための構文

| seq_parameter_set_rbsp( ) { | 記述子 | |
|---|---|---|
| … | | |
| **sps_log2_diff_min_qt_min_cb_intra_slice_luma** | ue(v) | |
| **sps_log2_diff_min_qt_min_cb_inter_slice** | ue(v) | |
| *sps_log2_diff_min_qt_min_cb_luma* | ue(v) | } 1401 |
| **sps_max_mtt_hierarchy_depth_inter_slice** | ue(v) | |
| **sps_max_mtt_hierarchy_depth_intra_slice_luma** | ue(v) | |
| *sps_max_mtt_depth_luma* | ue(v) | } 1402 |
| … | | |

【図１５】

表15: pps_max_mtt_depth_lumaのための構文

| pic_parameter_set_rbsp( ) { | 記述子 |
|---|---|
| … | |
| if( rect_slice_flag ) { | |
| **signalled_slice_id_flag** | u(1) |
| if( signalled_slice_id_flag ) { | |
| **signalled_slice_id_length_minus1** | ue(v) |
| for( i = 0; i <= num_slices_in_pic_minus1; i++ ) | |
| **slice_id**[ i ] | u(v) |
| } | |
| } | |
| *pps_log2_ctb_size* | *ue(v)* |
| *pps_log2_min_qt* | *ue(v)* |
| *pps_max_mtt_depth_luma* | *ue(v)* |
| … | |

1501

【図１６】

表16: cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivのためのSPS構文

| seq_parameter_set_rbsp( ) { | 記述子 |
|---|---|
| … | |
| **sps_max_mtt_hierarchy_depth_inter_slice** | ue(v) |
| **sps_max_mtt_hierarchy_depth_intra_slice_luma** | ue(v) |
| if( sps_max_mtt_hierarchy_depth_intra_slice_luma != 0 ) { | |
| **sps_log2_diff_max_bt_min_qt_intra_slice_luma** | ue(v) |
| **sps_log2_diff_max_tt_min_qt_intra_slice_luma** | ue(v) |
| } | |
| if( sps_max_mtt_hierarchy_depth_inter_slices != 0 ) { | |
| **sps_log2_diff_max_bt_min_qt_inter_slice** | ue(v) |
| **sps_log2_diff_max_tt_min_qt_inter_slice** | ue(v) |
| } | |
| *cu_qp_delta_enabled_flag* | *u(1)* |
| *if( cu_qp_delta_enabled_flag )* | |
| *sps_cu_qp_delta_subdiv* | *ue(v)* |
| *cu_chroma_qp_offset_enabled_flag* | *u(1)* |
| *if( cu_chroma_qp_offset_enabled_flag ) {* | |
| *sps_cu_chroma_qp_offset_subdiv* | *ue(v)* |
| … | |

1601

【図１７】

表17: cu_qp_delta_subdiv及びcu_chroma_qp_offset_subdivのためのスライスヘッダ構文

| slice_header( ) { | 記述子 |
|---|---|
| … | |
| if( partition_constraints_override_enabled_flag ) { | |
| **partition_constraints_override_flag** | ue(v) |
| if( partition_constraints_override_flag ) { | |
| **slice_log2_diff_min_qt_min_cb_luma** | ue(v) |
| **slice_max_mtt_hierarchy_depth_luma** | ue(v) |
| if( slice_max_mtt_hierarchy_depth_luma != 0 ){ | |
| **slice_log2_diff_max_bt_min_qt_luma** | ue(v) |
| **slice_log2_diff_max_tt_min_qt_luma** | ue(v) |
| } | |
| if( slice_type = = I && qtbtt_dual_tree_intra_flag ) { | |
| **slice_log2_diff_min_qt_min_cb_chroma** | ue(v) |
| **slice_max_mtt_hierarchy_depth_chroma** | ue(v) |
| if( slice_max_mtt_hierarchy_depth_chroma != 0 ){ | |
| **slice_log2_diff_max_bt_min_qt_chroma** | ue(v) |
| **slice_log2_diff_max_tt_min_qt_chroma** | ue(v) |
| } | |
| } | |
| } | |
| *if( cu_qp_delta_enabled_flag && partition_constraints_override_flag)* | |
| *slice_cu_qp_delta_subdiv* | *ue(v)* |
| *if( cu_chroma_qp_offset_enabled_flag && partition_constraints_override_flag) {* | |
| *slice_cu_chroma_qp_offset_subdiv* | *ue(v)* |
| | |

1701

【図１８】

表18: sps_max_mtt_depth_lumaのための構文

| seq_parameter_set_rbsp( ) { | 記述子 |
|---|---|
| … | |
| *sps_log2_diff_min_qt_min_cb_luma* | *ue(v)* |
| **sps_max_mtt_hierarchy_depth_inter_slice** | ue(v) |
| **sps_max_mtt_hierarchy_depth_intra_slice_luma** | ue(v) |
| *sps_max_mtt_depth_luma* | *ue(v)* |
| … | |

1801

1802

【図１９】

表19: sps_max_mtt_depth_lumaのための構文

| seq_parameter_set_rbsp( ) { | 記述子 |
|---|---|
| … | |
| **sps_max_mtt_hierarchy_depth_inter_slice** | ue(v) |
| **sps_max_mtt_hierarchy_depth_intra_slice_luma** | ue(v) |
| *sps_max_mtt_depth_luma* | *ue(v)* |
| … | |

1901

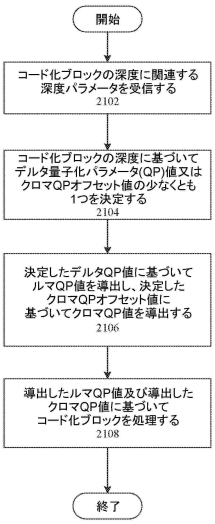
【図２０】

表20: pps_max_mtt_depth_lumaのための構文

| pic_parameter_set_rbsp( ) { | 記述子 |
|---|---|
| … | |
| if( rect_slice_flag ) { | |
| **signalled_slice_id_flag** | u(1) |
| if( signalled_slice_id_flag ) { | |
| **signalled_slice_id_length_minus1** | ue(v) |
| for( i = 0; i <= num_slices_in_pic_minus1; i++ ) | |
| **slice_id**[ i ] | u(v) |
| } | |
| } | |
| *pps_log2_ctb_size* | *ue(v)* |
| *pps_log2_min_qt* | *ue(v)* |
| *pps_max_mtt_depth_luma* | *ue(v)* |
| … | |

2001

【図２１】




2100

フロントページの続き

(51)国際特許分類　　ＦＩ
*Ｈ０４Ｎ　19/96　(2014.01)*　　Ｈ０４Ｎ　19/96

(72)発明者　チェン，ジエ
中華人民共和国，ジャージャン ３１１１２１，ハンチョウ，ユ ハン ディストリクト，ウェスト ウェン イ ロード ナンバー ９６９ ビルディング ３ ５／エフ，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
中華人民共和国，ジャージャン ３１１１２１，ハンチョウ，ユ ハン ディストリクト，ウェスト ウェン イ ロード ナンバー９６９ ビルディング ３ ５／エフ，アリババ グループ リーガル デパートメント
(72)発明者　ルオ，ジャンコン
アメリカ合衆国，カリフォルニア州 ９４４０２，サン マテオ，サウス エル カミーノ レアル ４００，スイート ４００，アリババ グループ リーガル デパートメント
(72)発明者　イエ，ヤン
アメリカ合衆国，カリフォルニア州 ９４４０２，サン マテオ，サウス エル カミーノ レアル ４００，スイート ４００，アリババ グループ リーガル デパートメント

審査官　田中 純一

(56)参考文献　特表２０２２－５４８４４８（ＪＰ，Ａ）
特開２０２１－０３４９６６（ＪＰ，Ａ）

(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｈ０４Ｎ　７／１２
Ｈ０４Ｎ　１９／００　－　１９／９８