(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7685499号
(P7685499)

(45)発行日　令和7年5月29日(2025.5.29)

(24)登録日　令和7年5月21日(2025.5.21)

(51)国際特許分類

		FI	
H04N	*19/119 (2014.01)*	H04N	19/119
H04N	*19/167 (2014.01)*	H04N	19/167
H04N	*19/70 (2014.01)*	H04N	19/70
H04N	*19/96 (2014.01)*	H04N	19/96

請求項の数　21　（全51頁）

(21)出願番号　特願2022-534413(P2022-534413)
(86)(22)出願日　令和2年11月24日(2020.11.24)
(65)公表番号　特表2023-507270(P2023-507270A)
(43)公表日　令和5年2月22日(2023.2.22)
(86)国際出願番号　PCT/US2020/061970
(87)国際公開番号　WO2021/126487
(87)国際公開日　令和3年6月24日(2021.6.24)
審査請求日　令和5年11月22日(2023.11.22)
(31)優先権主張番号　62/948,856
(32)優先日　令和1年12月17日(2019.12.17)
(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】　ピクチャ境界におけるブロック区分のための方法及び装置

(57)【特許請求の範囲】

【請求項１】

映像復号化方法であって、

対象符号化ユニットに関連付けられた第１のフラグの値を決定することであって、前記第１のフラグが、前記対象符号化ユニットが複数の符号化ユニットに分割されるかどうかを示すことと、

前記対象符号化ユニットに関連付けられた第１のパラメータ、第２のパラメータ、第３のパラメータ、第４のパラメータ、及び第５のパラメータの値を決定することであって、前記第１のパラメータが、前記対象符号化ユニットを分割するために、４分割を用いることが許可されるかどうかを示し、前記第２のパラメータが、前記対象符号化ユニットを分割するために、水平２分割を用いることが許可されるかどうかを示し、前記第３のパラメータが、前記対象符号化ユニットを分割するために、垂直２分割を用いることが許可されるかどうかを示し、前記第４のパラメータが、前記対象符号化ユニットを分割するために、水平３分割を用いることが許可されるかどうかを示し、前記第５のパラメータが、前記対象符号化ユニットを分割するために、垂直３分割を用いることが許可されるかどうかを示すことと、

前記複数の符号化ユニットの各々が、前記対象符号化ユニットの半分の水平サイズ及び半分の垂直サイズを有するかどうかを示す第２のフラグの値を決定することと、を含み、

前記第１のフラグが第２の値に等しく、且つ

前記第１のパラメータ、前記第２のパラメータ、前記第３のパラメータ、前記第４のパ

ラメータ、及び前記第５のパラメータが第３の値に等しいとき、
前記第２のフラグが第１の値に等しいと決定される、方法。
【請求項２】
前記第１の値が１であり、
前記第２の値が１であり、且つ
前記第３の値が０である、請求項１に記載の方法。
【請求項３】
前記第２のフラグが前記第１の値に等しいことが、前記複数の符号化ユニットの各々が、前記対象符号化ユニットの前記半分の水平サイズ及び前記半分の垂直サイズを有することを示す、請求項１に記載の方法。
【請求項４】
前記第１のフラグが前記第２の値に等しいことが、前記対象符号化ユニットが前記複数の符号化ユニットに分割されることを示す、請求項１に記載の方法。
【請求項５】
前記第１のパラメータ、前記第２のパラメータ、前記第３のパラメータ、前記第４のパラメータ、及び前記第５のパラメータが前記第３の値に等しいことが、前記４分割、水平２分割、垂直２分割、水平３分割、又は垂直３分割のいずれも、前記対象符号化ユニットを分割するために用いることを許可されないことを示す、請求項１に記載の方法。
【請求項６】
前記第１のパラメータが第４の値に等しいとき、前記第２のフラグが前記第１の値に等しいと決定される、請求項１に記載の方法。
【請求項７】
前記第１のパラメータが前記第４の値に等しいことが、前記対象符号化ユニットを分割するために前記４分割を用いることが許可されることを示す、請求項６に記載の方法。
【請求項８】
前記符号化ユニットがピクチャ境界の外側に１つ以上のサンプルを含むことに応じて、前記第１のフラグが前記第２の値に等しいと決定することをさらに含む、請求項１に記載の方法。
【請求項９】
前記第１のフラグの前記値を決定することが、
前記対象符号化ユニットに関連付けられたビットストリームから前記第１のフラグを復号化することを含む、請求項１に記載の方法。
【請求項１０】
前記第１のフラグがｓｐｌｉｔ＿ｃｕ＿ｆｌａｇを含み、前記第２のフラグがｓｐｌｉｔ＿ｑｔ＿ｆｌａｇを含む、請求項１に記載の方法。
【請求項１１】
前記対象符号化ユニットに関連付けられたビットストリームを受信することと、
前記第２のフラグが前記ビットストリームにおいてシグナリングされていないと決定することと、をさらに含む、請求項１に記載の方法。
【請求項１２】
映像符号化方法であって、
対象符号化ユニットがピクチャ境界の外側にサンプルを含むかどうかを決定することと、
前記対象符号化ユニットに関連付けられた第１のパラメータ、第２のパラメータ、第３のパラメータ、第４のパラメータ、及び第５のパラメータの値を決定することであって、
前記第１のパラメータが、前記対象符号化ユニットを分割するために、４分割を用いることが許可されるかどうかを示し、前記第２のパラメータが、前記対象符号化ユニットを分割するために、水平２分割を用いることが許可されるかどうかを示し、前記第３のパラメータが、前記対象符号化ユニットを分割するために、垂直２分割を用いることが許可されるかどうかを示し、前記第４のパラメータが、前記対象符号化ユニットを分割するために、水平３分割を用いることが許可されるかどうかを示し、前記第５のパラメータが、前記

対象符号化ユニットを分割するために、垂直３分割を用いることが許可されるかどうかを示すことと、
前記対象符号化ユニットが前記ピクチャ境界の外側にサンプルを含むこと、及び前記第１のパラメータ、前記第２のパラメータ、前記第３のパラメータ、前記第４のパラメータ、及び前記第５のパラメータが第１の値に等しいことに応じて、前記対象符号化ユニットを複数の符号化ユニットに分割することであって、前記複数の符号化ユニットの各々が、前記対象符号化ユニットの半分の水平サイズ及び半分の垂直サイズを有することと、
を含む、方法。
【請求項１３】
前記第１の値が０である、請求項１２に記載の方法。
【請求項１４】
前記第１のパラメータ、前記第２のパラメータ、前記第３のパラメータ、前記第４のパラメータ、及び前記第５のパラメータが前記第１の値に等しいことが、前記４分割、水平２分割、垂直２分割、水平３分割、又は垂直３分割のいずれも、前記対象符号化ユニットを分割するために用いることを許可されないことを示す、請求項１２に記載の方法。
【請求項１５】
映像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、
対象符号化ユニットがピクチャ境界の外側にサンプルを含むかどうかを決定することと、
前記対象符号化ユニットに関連付けられた第１のパラメータ、第２のパラメータ、第３のパラメータ、第４のパラメータ、及び第５のパラメータの値を決定することであって、前記第１のパラメータが、前記対象符号化ユニットを分割するために、４分割を用いることが許可されるかどうかを示し、前記第２のパラメータが、前記対象符号化ユニットを分割するために、水平２分割を用いることが許可されるかどうかを示し、前記第３のパラメータが、前記対象符号化ユニットを分割するために、垂直２分割を用いることが許可されるかどうかを示し、前記第４のパラメータが、前記対象符号化ユニットを分割するために、水平３分割を用いることが許可されるかどうかを示し、前記第５のパラメータが、前記対象符号化ユニットを分割するために、垂直３分割を用いることが許可されるかどうかを示すことと、
前記第１のパラメータ、前記第２のパラメータ、前記第３のパラメータ、前記第４のパラメータ、及び前記第５のパラメータに基づいて、前記対象符号化ユニットを表す符号化された情報を含むビットストリームを生成することと、
前記ビットストリームを非一時的コンピュータ可読記憶媒体に記憶することと、を含み、
前記対象符号化ユニットが前記ピクチャ境界の外側にサンプルを含み、かつ前記第１のパラメータ、前記第２のパラメータ、前記第３のパラメータ、前記第４のパラメータ、及び前記第５のパラメータが第１の値に等しい場合、前記対象符号化ユニットが複数の符号化ユニットに分割され、前記符号化された情報が前記複数の符号化ユニットに基づいて生成され、前記複数の符号化ユニットの各々が、前記対象符号化ユニットの半分の水平サイズ及び半分の垂直サイズを有する、方法。
【請求項１６】
前記第１の値が０である、請求項１５に記載の方法。
【請求項１７】
前記第１のパラメータ、前記第２のパラメータ、前記第３のパラメータ、前記第４のパラメータ、及び前記第５のパラメータが前記第１の値に等しいことが、前記４分割、水平２分割、垂直２分割、水平３分割、又は垂直３分割のいずれも、前記対象符号化ユニットを分割するために用いることを許可されないことを示す、請求項１５に記載の方法。
【請求項１８】
前記符号化された情報が、前記対象符号化ユニットが複数の符号化ユニットに分割されるかどうかを示す第１のフラグを含む、請求項１５に記載の方法。
【請求項１９】
前記符号化された情報が、前記複数の符号化ユニットの各々が前記対象符号化ユニットの

半分の水平サイズ及び半分の垂直サイズを有するかどうかを示す第２のフラグを含む、請求項１８に記載の方法。

【請求項２０】

前記第１のパラメータが、前記対象符号化ユニットを分割するために、前記４分割を用いることが許可されることを示す値を有し、前記第２のフラグが、複数の符号化ユニットの各々が、前記対象符号化ユニットの半分の水平サイズ及び半分の垂直サイズを有することを示す値に設定される、請求項１９に記載の方法。

【請求項２１】

前記第１のフラグがｓｐｌｉｔ＿ｃｕ＿ｆｌａｇを含み、前記第２のフラグがｓｐｌｉｔ＿ｑｔ＿ｆｌａｇを含む、請求項１９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、全体が本明細書において参照により組み込まれる、２０１９年１２月１７日に出願された、米国仮特許出願第６２／９４８，８５６号に対する優先権を主張する。

【０００２】

技術分野

［0002］ 本開示は、概して、映像処理に関し、より詳細には、ピクチャ境界においてブロック区分を実行するための方法及び装置に関する。

【背景技術】

【０００３】

背景

［0003］ 映像は、視覚情報を取り込んだ静的ピクチャ（又は「フレーム」）のセットである。記憶メモリ及び伝送帯域幅を低減するために、映像を記憶又は伝送前に圧縮し、表示前に復元することができる。圧縮プロセスは通常、符号化と称され、復元プロセスは通常、復号化と称される。最も一般的には、予測、変換、量子化、エントロピー符号化、及びインループフィルタリングに基づく、標準化映像符号化技術を用いる様々な映像符号化フォーマットが存在する。特定の映像符号化フォーマットを指定する、高効率ビデオコーディング（High Efficiency Video Coding）（ＨＥＶＣ／Ｈ．２６５）規格、多用途ビデオコーディング（Versatile Video Coding）（ＶＶＣ／Ｈ．２６６）、及び標準ＡＶＳ規格などの、映像符号化規格が標準化機関によって開発されている。進化した映像符号化技術が映像規格に次々と採用されるに従って、新たな映像符号化規格の符号化効率はますます高くなる。

【発明の概要】

【課題を解決するための手段】

【０００４】

開示の概要

［0004］ 実施形態によっては、例示的な映像処理方法は、符号化ブロックがピクチャ境界の外側にサンプルを含むかどうかを決定することと、符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されることに応じて、第１のパラメータの値に関わらず符号化ブロックの４分木分割を実行することであって、第１のパラメータが、符号化ブロックを分割するために４分木を用いることを許可されるかどうかを示すことと、を含む。

【０００５】

［0005］ 実施形態によっては、例示的な映像処理装置は、命令を記憶するための少なくとも１つのメモリと、少なくとも１つのプロセッサと、を含む。少なくとも１つのプロセッサは、符号化ブロックがピクチャ境界の外側にサンプルを含むかどうかを決定することと、符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されることに応じて、第１のパラメータの値に関わらず符号化ブロックの４分木分割を実行することであって、

第１のパラメータが、符号化ブロックを分割するために４分木を用いることを許可されるかどうかを示すことと、を装置に実行させるために、命令を実行するように構成される。
【０００６】
［0006］ 実施形態によっては、例示的な非一時的コンピュータ可読記憶媒体は、命令のセットを記憶する。命令のセットは、符号化ブロックがピクチャ境界の外側にサンプルを含むかどうかを決定することと、符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されることに応じて、第１のパラメータの値に関わらず符号化ブロックの４分木分割を実行することであって、第１のパラメータが、符号化ブロックを分割するために４分木を用いることを許可されるかどうかを示すことと、を映像処理装置に実行させるために、１つ以上の処理デバイスによって実行可能である。
【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が以下の詳細な説明及び添付の図面において例示される。図に示される様々な特徴は、原寸に比例して描かれていない。
【図面の簡単な説明】
【０００８】
【図１】[0008]図１は、本開示のいくつかの実施形態に係る、例示的な映像シーケンスの構造を示す概略図である。
【図２Ａ】[0009]図２Ａは、本開示の実施形態に従う、ハイブリッド映像符号化システムの例示的な符号化プロセスを示す概略図である。
【図２Ｂ】[0010]図２Ｂは、本開示の実施形態に従う、ハイブリッド映像符号化システムの別の例示的な符号化プロセスを示す概略図である。
【図３Ａ】[0011]図３Ａは、本開示の実施形態に従う、ハイブリッド映像符号化システムの例示的な復号化プロセスを示す概略図である。
【図３Ｂ】[0012]図３Ｂは、本開示の実施形態に従う、ハイブリッド映像符号化システムの別の例示的な復号化プロセスを示す概略図である。
【図４】[0013]図４は、本開示のいくつかの実施形態に係る、映像を符号化又は復号化するための例示的な装置のブロック図である。
【図５】[0014]本開示のいくつかの実施形態に係る、マルチタイプツリー分割モードの例を示す概略図である。
【図６】[0015]本開示のいくつかの実施形態に係る、ネスト型マルチタイプツリー符号化ツリー構造を有する４分木（ＱＴ）における区分分割情報のための例示的シグナリング機構を示す概略図である。
【図７】[0016]本開示のいくつかの実施形態に係る、マルチタイプツリー構文要素に基づく例示的なマルチタイプツリー分割モード（ＭｔｔＳｐｌｉｔＭｏｄｅ）導出を示す、例示的な表１を示す。
【図８】[0017]本開示のいくつかの実施形態に係る、許可されない３分木（ＴＴ）及び２分木（ＢＴ）区分の例を示す概略図である。
【図９】[0018]本開示のいくつかの実施形態に係る、ピクチャ境界上の例示的ブロック区分を示す概略図である。
【図１０】[0019]本開示のいくつかの実施形態に係る、２分割モード（ｂｔＳｐｌｉｔ）に基づく平行３分木分割（ｐａｒａｌｌｅｌＴｔＳｐｌｉｔ）及び符号化ブロックサイズ（ｃｂＳｉｚｅ）の例示的仕様を示す例示的な表２を示す。
【図１１】[0020]本開示のいくつかの実施形態に係る、３分木分割モード（ｔｔＳｐｌｉｔ）に基づくｃｂＳｉｚｅの例示的仕様を示す例示的な表３を示す。
【図１２Ａ】[0021]本開示のいくつかの実施形態に係る、例示的な符号化ツリー構文を示す例示的な表４を示す。
【図１２Ｂ】[0021]本開示のいくつかの実施形態に係る、例示的な符号化ツリー構文を示す例示的な表４を示す。
【図１２Ｃ】[0021]本開示のいくつかの実施形態に係る、例示的な符号化ツリー構文を示

す例示的な表４を示す。
【図１３】[0022]本開示のいくつかの実施形態に係る、ＭｔｔＳｐｌｉｔＭｏｄｅによって示されるマルチタイプツリー分割モードの例を示す概略図である。
【図１４】[0023]本開示のいくつかの実施形態に係る、ＭｔｔＳｐｌｉｔＭｏｄｅの例示的仕様を示す例示的な表５を示す。
【図１５】[0024]本開示のいくつかの実施形態に係る、例示的なシーケンスパラメータセットＲＢＳＰ構文を示す例示的な表６を示す。
【図１６】[0025]本開示のいくつかの実施形態に係る、例示的なピクチャヘッダＲＢＳＰ構文を示す例示的な表７を示す。
【図１７】[0026]本開示のいくつかの実施形態に係る、ＱＴ、ＴＴ、又はＢＴ分割のいずれもピクチャ境界で許可されない例示的ブロックを示す概略図である。
【図１８】[0027]本開示のいくつかの実施形態に係る、ＱＴ、ＴＴ、又はＢＴ分割のいずれもピクチャ境界で許可されない例示的ブロックを示す概略図である。
【図１９】[0028]本開示のいくつかの実施形態に係る、ＢＴ及びＴＴ分割を用いる例を示す概略図である。
【図２０】[0029]本開示のいくつかの実施形態に係る、例示的映像処理方法のフローチャートを示す。
【発明を実施するための形態】
【０００９】
詳細な説明
[0030] 次に、添付の図面に例が示された、例示的な実施形態を詳細に参照する。以下の説明は添付の図面を参照し、図面において、異なる図面における同じ符号は、別途示されない限り、同じ又は同様の要素を表す。例示的な実施形態の以下の説明において示される実装形態は、本発明に従う全ての実装形態を表すものではない。むしろ、それらは、添付の請求項において列挙されるとおりの本発明に関連する態様に従う装置及び方法の単なる例にすぎない。本開示の特定の態様が以下においてより詳細に説明される。参照により組み込まれる用語及び／又は定義と矛盾する場合には、本明細書において提供される用語及び定義が優先する。
【００１０】
[0031] ＩＴＵ－Ｔビデオコーディングエキスパートグループ（ＩＴＵ－Ｔ ＶＣＥＧ（ITU-T Video Coding Expert Group））及びＩＳＯ／ＩＥＣムービングピクチャエクスパートグループ（ＩＳＯ／ＩＥＣＭＰＥＧ（ISO/IEC Moving Picture Expert））のジョイントビデオエクスパーツチーム（ＪＶＥＴ（Joint Video Experts Team））は、現在、多用途ビデオコーディング（ＶＶＣ／Ｈ．２６６）規格を開発している。ＶＶＣ規格は、その前身、高効率ビデオコーディング（ＨＥＶＣ／Ｈ．２６５）規格の圧縮効率を２倍にすることを目指している。換言すれば、ＶＶＣの目標は、半分の帯域幅を用いてＨＥＶＣ／Ｈ．２６５と同じ主観的品質を達成することである。
【００１１】
[0032] 半分の帯域幅を用いてＨＥＶＣ／Ｈ．２６５と同じ主観的品質を達成するために、ＪＶＥＴは、共同探索モデル（ＪＥＭ（joint exploration model））参照ソフトウェアを用いてＨＥＶＣを超える技術を開発している。符号化技術がＪＥＭに組み込まれたため、ＪＥＭはＨＥＶＣよりも実質的に高い符号化性能を達成した。
【００１２】
[0033] ＶＶＣ規格は最近開発されたものであり、より優れた圧縮性能をもたらすより多くの符号化技術を組み込み続けている。ＶＶＣは、ＨＥＶＣ、Ｈ．２６４／ＡＶＣ、MPEG2、H.263等などの現代的な映像圧縮規格において用いられてきた同じハイブリッド映像符号化システムに基づく。
【００１３】
[0034] 映像は、視覚情報を記憶するために時系列で配列された静的ピクチャ（又は「フレーム」）のセットである。映像取り込みデバイス（例えば、カメラ）を、それらのピ

クチャを時系列で取り込んで記憶するために用いることができ、映像再生デバイス（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、このようなピクチャを時系列で表示するために用いることができる。また、用途によっては、映像取り込みデバイスが、取り込まれた映像を、監督、会議開催、又は生放送などのために、映像再生デバイス（例えば、モニタを有するコンピュータ）へリアルタイムに伝送することができる。

【００１４】

［0035］ このような用途によって必要とされる記憶空間及び伝送帯域幅を低減するために、映像を記憶及び伝送前に圧縮し、表示前に復元することができる。圧縮及び復元は、プロセッサ（例えば、汎用コンピュータのプロセッサ）によって実行されるソフトウェア、又は特殊ハードウェアによって実施され得る。圧縮のためのモジュールは一般的に「符号器」と称され、復元のためのモジュールは一般的に「復号器」と称される。符号器及び復号器はまとめて「コーデック」と称され得る。符号器及び復号器は、種々の好適なハードウェア、ソフトウェア、又はこれらの組み合わせのうちの任意のものとして実施することができる。例えば、符号器及び復号器のハードウェア実装形態は、１つ以上のマイクロプロセッサ、デジタル信号プロセッサ（DSP（digital signal processor））、特定用途向け集積回路（ASIC（application-specific integrated circuit））、フィールドプログラマブルゲートアレイ（FPGA（field-programmable gate array））、個別論理、又はこれらの任意の組み合わせなどの、回路機構を含むことができる。符号器及び復号器のソフトウェア実装形態は、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た任意の好適なコンピュータ実施アルゴリズム若しくはプロセスを含むことができる。映像圧縮及び復元は、ＭＰＥＧ－１、ＭＰＥＧ－２、ＭＰＥＧ－４、Ｈ．２６ｘシリーズ、又は同様のものなどの、様々なアルゴリズム又は規格によって実施され得る。用途によっては、コーデックは映像を第１の符号化規格から復元し、復元された映像を、第２の符号化規格を用いて再圧縮することができる。この場合には、コーデックは「トランスコーダ」と称され得る。

【００１５】

［0036］ 映像符号化プロセスは、ピクチャを再構成するために用いることができる有用な情報を識別して維持し、再構成のため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成することができない場合には、このような符号化プロセスは「非可逆」と称され得る。さもなければ、それは「可逆」と称され得る。大抵の符号化プロセスは非可逆であり、これは、必要とされる記憶空間及び伝送帯域幅を低減するためのトレードオフである。

【００１６】

［0037］ 符号化中のピクチャ（「現在のピクチャ」と称される）の有用な情報は、参照ピクチャ（例えば、以前に符号化され、再構成されたピクチャ）に対する変化を含む。このような変化は、ピクセルの位置の変化、明るさの変化、又は色の変化を含むことができ、これらの中でも、位置の変化に最も重要である。オブジェクトを表現するピクセルのグループの位置の変化は、参照ピクチャと現在のピクチャとの間のオブジェクトの動きを反映することができる。

【００１７】

［0038］ 別のピクチャを参照することなく符号化されたピクチャ（すなわち、それがそれ自身の参照ピクチャである）は「Ｉピクチャ」と称される。以前のピクチャを参照ピクチャとして用いて符号化されたピクチャは「Ｐピクチャ」と称される。以前のピクチャ及び将来のピクチャの両方を参照ピクチャとして用いて符号化されたピクチャ（すなわち、参照は「双方向性」である）は「Ｂピクチャ」と称される。

【００１８】

［0039］ 図１は、本開示のいくつかの実施形態に係る、例示的な映像シーケンス１００の構造を示す。映像シーケンス１００は、ライブ映像、又は取り込まれ、アーカイブされた映像であることができる。映像１００は、現実の映像、コンピュータ生成映像（例えば

、コンピュータゲーム映像）、又はこれらの組み合わせ（例えば、拡張現実感効果を伴う現実の映像）であることができる。映像シーケンス１００は、映像取り込みデバイス（例えば、カメラ）、以前に取り込まれた映像を包含する映像アーカイブ（例えば、記憶デバイス内に記憶された映像ファイル）、又は映像コンテンツプロバイダからの映像を受信するための映像供給インターフェース（例えば、映像放送トランシーバ）から入力され得る。

【００１９】

［0040］図１に示されるように、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列された一連のピクチャを含むことができる。ピクチャ１０２～１０６は連続しており、ピクチャ１０６及び１０８の間にはさらなるピクチャが存在する。図１において、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自身である。ピクチャ１０４はＰピクチャであり、その参照ピクチャは、矢印によって指示されるように、ピクチャ１０２である。ピクチャ１０６はＢピクチャであり、その参照ピクチャは、矢印によって指示されるように、ピクチャ１０４及び１０８である。実施形態によっては、ピクチャ（例えば、ピクチャ１０４）の参照ピクチャは当該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の前のピクチャであることができる。ピクチャ１０２～１０６の参照ピクチャは単なる例にすぎず、本開示は参照ピクチャの実施形態を、図１に示される例として限定しないことに留意されたい。

【００２０】

［0041］典型的に、映像コーデックは、このようなタスクの計算の複雑性のゆえに、ピクチャ全体を一度に符号化又は復号化しない。むしろ、それら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化することができる。このような基本セグメントは本開示において基本処理ユニット（「ＢＰＵ（basic processing unit）」）と称される。例えば、図１における構造１１０は、映像シーケンス１００のピクチャ（例えば、ピクチャ１０２～１０８のうちの任意のもの）の例示的な構造を示す。構造１１０においては、ピクチャが４×４基本処理ユニットに分割され、それらの境界は破線として示されている。実施形態によっては、基本処理ユニットは、いくつかの映像符号化規格（例えば、MPEGファミリー、Ｈ．２６１、Ｈ．２６３、又はＨ．２６４／ＡＶＣ）では、「マクロブロック」と、或いはいくつかの他の映像符号化規格（例えば、Ｈ．２６５／ＨＥＶＣ又はＨ．２６６／ＶＶＣ）では、「符号化ツリーユニット」（「ＣＴＵ（coding tree unit）」）と称され得る。基本処理ユニットは、１２８×１２８、６４×６４、３２×３２、１６×１６、４×８、１６×３２などの、ピクチャにおける可変サイズ、或いはピクセルの任意の形状及びサイズを有することができる。基本処理ユニットのサイズ及び形状は、ピクチャのために、符号化効率と、基本処理ユニットにおいて維持されるべき詳細さのレベルとのバランスに基づいて選択することができる。

【００２１】

［0042］基本処理ユニットは、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）記憶された異なる種類の映像データのグループを含むことができる、論理ユニットであることができる。例えば、カラーピクチャの基本処理ユニットは、無色の輝度情報を表現するルマ成分（Ｙ）、色情報を表現する１つ以上のクロマ成分（例えば、Ｃｂ及びＣｒ）、並びに関連構文要素を含むことができ、ここで、ルマ及びクロマ成分は基本処理ユニットの同じサイズを有することができる。ルマ及びクロマ成分は、いくつかの映像符号化規格（例えば、Ｈ．２６５／ＨＥＶＣ又はＨ．２６６／ＶＶＣ）では、「符号化ツリーブロック」（「ＣＴＢ（coding tree block）」）と称され得る。基本処理ユニットに対して遂行される任意の演算は、そのルマ及びクロマ成分の各々に対して繰り返し遂行され得る。

【００２２】

［0043］映像符号化は複数の演算段階を有し、図２Ａ～図２Ｂ及び図３Ａ～図３Ｂにその例が示されている。段階ごとに、基本処理ユニットのサイズは依然として処理のために大きすぎるものになり得、それゆえ、本開示において「基本処理サブユニット」と称され

るセグメントにさらに分割され得る。実施形態によっては、基本処理サブユニットは、いくつかの映像符号化規格（例えば、ＭＰＥＧファミリー、Ｈ．２６１、Ｈ．２６３、又はＨ．２６４／ＡＶＣ）では、「ブロック」と、或いはいくつかの他の映像符号化規格（例えば、Ｈ．２６５／ＨＥＶＣ又はＨ．２６６／ＶＶＣ）では、「符号化ユニット」（「ＣＵ」）と称され得る。基本処理サブユニットは、基本処理ユニットと同じ、又はそれよりも小さいサイズを有することができる。基本処理ユニットと同様に、基本処理サブユニットもまた、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）記憶された異なる種類の映像データ（例えば、Ｙ、Ｃｂ、Ｃｒ、及び関連構文要素）のグループを含むことができる、論理ユニットである。基本処理サブユニットに対して遂行される任意の動作は、そのルマ及びクロマ成分の各々に対して繰り返し遂行され得る。このような分割は処理の必要に応じてさらなるレベルまで遂行され得ることに留意されたい。また、異なる段階は、異なる方式を用いて基本処理ユニットを分割することができることにも留意されたい。

【００２３】

［0044］ 例えば、モード決定段階（図２Ｂにその一例が示されている）において、符号器は、どのような予測モード（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）を基本処理ユニットのために用いるかを決定することができるが、基本処理ユニットは、このような決定を行うには大きすぎるものになり得る。符号器は、基本処理ユニットを複数の基本処理サブユニット（例えば、Ｈ．２６５／ＨＥＶＣ又はＨ．２６６／ＶＶＣの場合のように、ＣＵ）に分割し、個々の基本処理サブユニットごとに予測の種類を決めることができる。

【００２４】

［0045］ 別の例として、予測段階（図２Ａ～図２Ｂにその例が示されている）において、符号器は、基本処理サブユニット（例えば、ＣＵ）のレベルで予測演算を遂行することができる。しかし、場合によっては、基本処理サブユニットは、依然として、処理するには大きすぎるものになり得る。符号器は、基本処理サブユニットをより小さいセグメント（例えば、Ｈ．２６５／ＨＥＶＣ又はＨ．２６６／ＶＶＣでは「予測ブロック（prediction block）」又は「ＰＢ」と称される）にさらに分割することができ、そのレベルで予測演算が遂行され得る。

【００２５】

［0046］ 別の例として、変換段階（図２Ａ～図２Ｂにその例が示されている）において、符号器は、残差基本処理サブユニット（例えば、ＣＵ）のための変換演算を遂行することができる。しかし、場合によっては、基本処理サブユニットは、依然として、処理するには大きすぎるものになり得る。符号器は、基本処理サブユニットをより小さいセグメント（例えば、Ｈ．２６５／ＨＥＶＣ又はＨ．２６６／ＶＶＣでは「変換ブロック（transform block）」又は「ＴＢ」と称される）にさらに分割することができ、そのレベルで変換演算が遂行され得る。同じ基本処理サブユニットの分割方式が、予測段階及び変換段階において異なってもよいことに留意されたい。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が異なるサイズ及び数を有してもよい。

【００２６】

［0047］ 図１の構造１１０においては、基本処理ユニット１１２は、３×３基本処理サブユニットにさらに分割され、それらの境界は点線として示されている。同じピクチャの異なる基本処理ユニットは異なる方式で基本処理サブユニットに分割されてもよい。

【００２７】

［0048］ 実装形態によっては、並列処理及び誤り耐性の能力を映像符号化及び復号化にもたらすために、ピクチャを処理のための領域に分割することができ、これにより、符号化又は復号化プロセスは、ピクチャの領域に関して、ピクチャのいかなる他の領域からの情報にも依存しなくてすむ。換言すれば、ピクチャの各領域は独立して処理され得る。そうすることによって、コーデックはピクチャの異なる領域を並行して処理することができ

、それゆえ、符号化効率を増大させる。また、領域のデータが処理中に破損したか、又はネットワーク伝送中に失われたときには、コーデックは、破損した、又は失われたデータを頼ることなく、同じピクチャの他の領域を正しく符号化又は復号化することもでき、それゆえ、誤り耐性の能力をもたらす。いくつかの映像符号化規格では、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２種類の領域：「スライス」及び「タイル」を提供する。また、映像シーケンス１００の異なるピクチャは、ピクチャを領域に分割するための異なる区分方式を有することができることにも留意されたい。

【００２８】

［0049］ 例えば、図１では、構造１１０は、３つの領域１１４、１１６、及び１１８に分割され、それらの境界は構造１１０の内部の実線として示されている。領域１１４は４つの基本処理ユニットを含む。領域１１６及び１１８の各々は６つの基本処理ユニットを含む。図１における構造１１０の基本処理ユニット、基本処理サブユニット、及び領域は単なる例に過ぎず、本開示はその実施形態を限定しないことに留意されたい。

【００２９】

［0050］ 図２Ａは、本開示の実施形態に従う、例示的な符号化プロセス２００Ａの概略図を示す。例えば、符号化プロセス２００Ａは符号器によって遂行され得る。図２Ａに示されるように、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における映像シーケンス１００と同様に、映像シーケンス２０２は、時間的順序で配列されたピクチャ（「原ピクチャ」と称される）のセットを含むことができる。図１における構造１１０と同様に、映像シーケンス２０２の各原ピクチャは、符号器によって処理のために基本処理ユニット、基本処理サブユニット、又は領域に分割され得る。実施形態によっては、符号器は、映像シーケンス２０２の原ピクチャごとに基本処理ユニットのレベルでプロセス２００Ａを遂行することができる。例えば、符号器はプロセス２００Ａを反復的な仕方で遂行することができ、その場合、符号器は基本処理ユニットをプロセス２００Ａの１回の反復において符号化することができる。実施形態によっては、符号器は、プロセス２００Ａを映像シーケンス２０２の各原ピクチャの領域（例えば、領域１１４～１１８）のために並行して遂行することができる。

【００３０】

［0051］ 図２Ａにおいて、符号器は、映像シーケンス２０２の原ピクチャの基本処理ユニット（「原ＢＰＵ」と称される）を予測段階２０４に供給し、予測データ２０６及び予測ＢＰＵ２０８を生成することができる。符号器は、予測ＢＰＵ２０８を原ＢＰＵから減算し、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に供給し、量子化変換係数２１６を生成することができる。符号器は、予測データ２０６及び量子化変換係数２１６を２値符号化段階２２６に供給し、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」と称され得る。プロセス２００Ａの間、量子化段階２１４の後に、符号器は、量子化変換係数２１６を逆量子化段階２１８及び逆変換段階２２０に供給し、再構成残差ＢＰＵ２２２を生成することができる。符号器は、再構成残差ＢＰＵ２２２を予測ＢＰＵ２０８に加算し、プロセス２００Ａの次の反復のために予測段階２０４において用いられる、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「再構成経路」と称され得る。再構成経路は、符号器及び復号器の両方が同じ参照データを予測のために用いることを確実にするために用いられ得る。

【００３１】

［0052］ 符号器は、原ピクチャの各原ＢＰＵを（順方向経路内で）符号化し、原ピクチャの次の原ＢＰＵを符号化するための予測基準２２４を（再構成経路内で）生成するために、プロセス２００Ａを反復的に遂行することができる。原ピクチャの全ての原ＢＰＵを符号化した後に、符号器は、映像シーケンス２０２内の次のピクチャを符号化するために

進むことができる。
【００３２】
［0053］プロセス２００Ａを参照すると、符号器は、映像取り込みデバイス（例えば、カメラ）によって生成された映像シーケンス２０２を受信することができる。本明細書において用いられる用語「受信する」は、受信すること、入力すること、獲得すること、取得すること、得ること、読み込むこと、アクセスすること、又はデータを入力するための任意の仕方による任意の行為を指すことができる。
【００３３】
［0054］予測段階２０４において、現在の反復において、符号器は原ＢＰＵ及び予測基準２２４を受信し、予測演算を遂行し、予測データ２０６及び予測ＢＰＵ２０８を生成することができる。予測基準２２４は、プロセス２００Ａの以前の反復の再構成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って、情報冗長性を低減することであり、予測データ２０６は、予測データ２０６及び予測基準２２４から原ＢＰＵを予測ＢＰＵ２０８として再構成するために用いることができる。
【００３４】
［0055］理想的には、予測ＢＰＵ２０８は原ＢＰＵと同一であることができる。しかし、非理想的な予測及び再構成演算のゆえに、予測ＢＰＵ２０８は、概して、原ＢＰＵとは若干異なる。このような差を記録するために、予測ＢＰＵ２０８を生成した後に、符号器はそれを原ＢＰＵから減算し、残差ＢＰＵ２１０を生成することができる。例えば、符号器は、予測ＢＰＵ２０８のピクセルの値（例えば、グレースケール値又はＲＧＢ値）を原ＢＰＵの対応するピクセルの値から減算することができる。残差ＢＰＵ２１０の各ピクセルは、原ＢＰＵ及び予測ＢＰＵ２０８の対応するピクセルの間のこのような減算の結果としての残差値を有することができる。原ＢＰＵと比べて、予測データ２０６及び残差ＢＰＵ２１０はより少数のビットを有することができるが、それらは、著しい品質劣化を伴うことなく原ＢＰＵを再構成するために用いられ得る。それゆえ、原ＢＰＵは圧縮される。
【００３５】
［0056］残差ＢＰＵ２１０をさらに圧縮するために、変換段階２１２において、符号器は、それを２次元「基底パターン」のセットに分解することによって、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」に関連付けられている。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。各基底パターンは残差ＢＰＵ２１０の変化周波数（例えば、輝度変化の周波数）成分を表現することができる。基底パターンはいずれも、いかなる他の基底パターンのいかなる結合（例えば、線形結合）からも再現することができない。換言すれば、分解は残差ＢＰＵ２１０の変化を周波数領域に分解することができる。このような分解は関数の離散フーリエ変換と類似しており、この場合、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）と類似しており、変換係数は、基底関数に関連付けられた係数と類似している。
【００３６】
［0057］異なる変換アルゴリズムは異なる基底パターンを用いることができる。例えば、離散余弦変換、離散正弦変換、又は同様のものなどの、様々な変換アルゴリズムを変換段階２１２において用いることができる。変換段階２１２における変換は逆演算可能である。すなわち、符号器は、変換の逆演算（「逆変換」と称される）によって残差ＢＰＵ２１０を回復することができる。例えば、残差ＢＰＵ２１０のピクセルを回復するために、逆変換は、基底パターンの対応するピクセルの値にそれぞれの関連係数を乗算し、積を加算していき、加重和を生成することができる。映像符号化規格のために、符号器及び復号器は両方とも同じ変換アルゴリズム（それゆえ、同じ基底パターン）を用いることができる。それゆえ、符号器は変換係数のみを記録することができ、復号器は、基底パターンを符号器から受信することなく、変換係数から残差ＢＰＵ２１０を再構成することができる。残差ＢＰＵ２１０と比べて、変換係数はより少数のビットを有することができるが、それらは、著しい品質劣化を伴うことなく残差ＢＰＵ２１０を再構成するために用いられ得

る。それゆえ、残差ＢＰＵ２１０はさらに圧縮される。

【００３７】

［0058］符号器は、量子化段階２１４において変換係数をさらに圧縮することができる。変換プロセスにおいて、異なる基底パターンは異なる変化周波数（例えば、輝度変化周波数）を表現することができる。人間の眼は、概して、低周波数変化を認識することがより得意であるため、符号器は、復号化において著しい品質劣化を生じさせることなく高周波数変化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」と称される）で除算し、商をその最近傍の整数に丸めることによって、量子化変換係数２１６を生成することができる。このような演算の後に、高周波数基底パターンの一部の変換係数は０に変換され得、低周波数基底パターンの変換係数はより小さい整数に変換され得る。符号器は０値の量子化変換係数２１６を無視することができ、これによって変換係数はさらに圧縮される。量子化プロセスもまた逆演算可能であり、この場合、量子化変換係数２１６は量子化の逆演算（「逆量子化」と称される）において変換係数に再構成され得る。

【００３８】

［0059］符号器はこのような除算の剰余を丸め演算において無視するため、量子化段階２１４は非可逆になり得る。典型的に、量子化段階２１４はプロセス２００Ａにおいて最大の情報損失に寄与し得る。情報損失が大きいほど、量子化変換係数２１６に必要なビットは少なくなる。異なる情報損失レベルを得るために、符号器は、量子化パラメータ又は量子化プロセスの任意の他のパラメータの異なる値を用いることができる。

【００３９】

［0060］２値符号化段階２２６において、符号器は、例えば、エントロピー符号化、可変長符号化、算術符号化、ハフマン符号化、コンテキスト適応２値算術符号化、又は任意の他の可逆若しくは非可逆圧縮アルゴリズムなどの、２値符号化技法を用いて、予測データ２０６及び量子化変換係数２１６を符号化することができる。実施形態によっては、予測データ２０６及び量子化変換係数２１６のほかに、符号器は、例えば、予測段階２０４において用いられる予測モード、予測演算のパラメータ、変換段階２１２における変換の種類、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）、又は同様のものなどの、他の情報を２値符号化段階２２６において符号化することができる。符号器は、２値符号化段階２２６の出力データを用いて映像ビットストリーム２２８を生成することができる。実施形態によっては、映像ビットストリーム２２８をネットワーク伝送のためにさらにパケット化することができる。

【００４０】

［0061］プロセス２００Ａの再構成経路を参照すると、逆量子化段階２１８において、符号器は、量子化変換係数２１６に対して逆量子化を遂行し、再構成変換係数を生成することができる。逆変換段階２２０において、符号器は、再構成変換係数に基づいて再構成残差ＢＰＵ２２２を生成することができる。符号器は、再構成残差ＢＰＵ２２２を予測ＢＰＵ２０８に加算し、プロセス２００Ａの次の反復において用いられることになる予測基準２２４を生成することができる。

【００４１】

［0062］プロセス２００Ａの他の変形を、映像シーケンス２０２を符号化するために用いることもできることに留意されたい。実施形態によっては、プロセス２００Ａの段階は符号器によって、異なる順序で遂行され得る。実施形態によっては、プロセス２００Ａの１つ以上の段階を単一の段階に組み合わせることができる。実施形態によっては、プロセス２００Ａの単一の段階を複数の段階に分割することができる。例えば、変換段階２１２及び量子化段階２１４を単一の段階に組み合わせることができる。実施形態によっては、プロセス２００Ａは追加の段階を含むことができる。実施形態によっては、プロセス２００Ａは図２Ａにおける１つ以上の段階を省略することができる。

【００４２】

［0063］図２Ｂは、本開示の実施形態に従う、別の例示的な符号化プロセス２００Ｂの概略図を示す。プロセス２００Ｂはプロセス２００Ａから変更され得る。例えば、プロセス２００Ｂは、ハイブリッド映像符号化規格（例えば、Ｈ．２６ｘシリーズ）に準拠した符号器によって用いられ得る。プロセス２００Ａと比べて、プロセス２００Ｂの順方向経路はモード決定段階２３０を追加的に含み、予測段階２０４を空間的予測段階２０４２及び時間的予測段階２０４４に分割する。プロセス２００Ｂの再構成経路はループフィルタ段階２３２及びバッファ２３４を追加的に含む。

【００４３】

［0064］概して、予測技法は２つの種類：空間的予測及び時間的予測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、現在のＢＰＵを予測するために、同じピクチャ内の１つ以上のすでに符号化された隣接ＢＰＵからのピクセルを用いることができる。すなわち、空間的予測における予測基準２２４は隣接ＢＰＵを含むことができる。空間的予測はピクチャの固有の空間的冗長性を低減することができる。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、現在のＢＰＵを予測するために、１つ以上のすでに符号化されたピクチャからの領域を用いることができる。すなわち、時間的予測における予測基準２２４は符号化ピクチャを含むことができる。時間的予測はピクチャの固有の時間的冗長性を低減することができる。

【００４４】

［0065］プロセス２００Ｂを参照すると、順方向経路内において、符号器は、予測演算を空間的予測段階２０４２及び時間的予測段階２０４４において遂行する。例えば、空間的予測段階２０４２において、符号器はイントラ予測を遂行することができる。符号化中のピクチャの原ＢＰＵのために、予測基準２２４は、（順方向経路内で）符号化され、（再構成経路内で）再構成された１つ以上の隣接ＢＰＵを同じピクチャ内に含むことができる。符号器は、隣接ＢＰＵを外挿することによって予測ＢＰＵ２０８を生成することができる。外挿技法は、例えば、線形外挿若しくは補間、多項式外挿若しくは補間、又は同様のものを含むことができる。実施形態によっては、符号器は、予測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るなどして、外挿をピクセルレベルで遂行することができる。外挿のために用いられる隣接ＢＰＵは、（例えば、原ＢＰＵの上の）鉛直方向、（例えば、原ＢＰＵの左の）水平方向、（例えば、原ＢＰＵの左下、右下、左上、若しくは右上の）対角方向、又は用いられる映像符号化規格において定義される任意の方向などの、様々な方向から原ＢＰＵに対して位置することができる。イントラ予測のために、予測データ２０６は、例えば、用いられる隣接ＢＰＵの場所（例えば、座標）、用いられる隣接ＢＰＵのサイズ、外挿のパラメータ、原ＢＰＵに対する用いられる隣接ＢＰＵの方向、又は同様のものを含むことができる。

【００４５】

［0066］別の例として、時間的予測段階２０４４において、符号器はインター予測を遂行することができる。現在のピクチャの原ＢＰＵのために、予測基準２２４は、（順方向経路内で）符号化され、（再構成経路内で）再構成された１つ以上のピクチャ（「参照ピクチャ」と称される）を含むことができる。実施形態によっては、参照ピクチャはＢＰＵごとに符号化され、再構成され得る。例えば、符号器は、再構成残差ＢＰＵ２２２を予測ＢＰＵ２０８に加算し、再構成ＢＰＵを生成することができる。同じピクチャの全ての再構成ＢＰＵが生成されたとき、符号器は再構成ピクチャを参照ピクチャとして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」と称される）内のマッチング領域を探索するために「動き推定」の演算を遂行することができる。参照ピクチャ内の探索窓の場所は、現在のピクチャの原ＢＰＵの場所に基づいて決定することができる。例えば、探索窓は、参照ピクチャ内の、現在のピクチャ内の原ＢＰＵと同じ座標を有する場所に中心を有することができ、所定の距離にわたって外へ拡張され得る。符号器が（例えば、画素再帰（pel-recursive）アルゴリズム、ブロックマッチングアルゴリズム、又は同様のものを用いることによって）探索窓内の原ＢＰＵと同様の領域を識別したとき、符号器はこのような領域をマッチング領域と決定することができる。マッチング領域は、原ＢＰ

Ｕとは異なる（例えば、原ＢＰＵよりも小さい、それに等しい、それよりも大きい、又は異なる形状の）寸法を有することができる。参照ピクチャ及び現在のピクチャが（例えば、図１に示されるように）タイムライン内で時間的に分離されているため、時間が経過するにつれてマッチング領域は原ＢＰＵの場所へ「移動する」と見なすことができる。符号器はこのような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。複数の参照ピクチャが（例えば、図１におけるピクチャ１０６として）用いられるときには、符号器は、参照ピクチャごとにマッチング領域を探索し、その関連動きベクトルを決定することができる。実施形態によっては、符号器は、それぞれのマッチング参照ピクチャのマッチング領域のピクセル値に重みを付与することができる。

【００４６】

［0067］動き推定は、例えば、並進、回転、ズーミング、又は同様のものなどの、様々な種類の動きを識別するために用いることができる。インター予測のために、予測データ２０６は、例えば、マッチング領域の場所（例えば、座標）、マッチング領域に関連付けられた動きベクトル、参照ピクチャの数、参照ピクチャに関連付けられた重み、又は同様のものを含むことができる。

【００４７】

［0068］予測ＢＰＵ２０８を生成するために、符号器は「動き補償」の演算を遂行することができる。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ＢＰＵ２０８を再構成するために用いることができる。例えば、符号器は、動きベクトルに従って参照ピクチャのマッチング領域を移動させることができ、その場合、符号器は現在のピクチャの原ＢＰＵを予測することができる。複数の参照ピクチャが（例えば、図１におけるピクチャ１０６として）用いられるときには、符号器は、それぞれの動きベクトルに従って参照ピクチャのマッチング領域を移動させ、マッチング領域のピクセル値を平均することができる。実施形態によっては、符号器がそれぞれのマッチング参照ピクチャのマッチング領域のピクセル値に重みを付与した場合には、符号器はピクセル値の加重和を、移動させられたマッチング領域に加算することができる。

【００４８】

［0069］実施形態によっては、インター予測は一方向性又は双方向性であることができる。一方向性インター予測は現在のピクチャに対して同じ時間方向の１つ以上の参照ピクチャを用いることができる。例えば、図１におけるピクチャ１０４は、参照ピクチャ（すなわち、ピクチャ１０２）がピクチャ１０４に先行する、一方向インター予測ピクチャである。双方向インター予測は、現在のピクチャに対して両方の時間方向にある１つ以上の参照ピクチャを用いることができる。例えば、図１におけるピクチャ１０６は、参照ピクチャ（すなわち、ピクチャ１０４及び１０８）がピクチャ１０４に対して両方の時間方向にある、双方向インター予測ピクチャである。

【００４９】

［0070］プロセス２００Ｂの順方向経路をなおも参照すると、空間的予測２０４２及び時間的予測段階２０４４の後に、モード決定段階２３０において、符号器は、予測モード（例えば、イントラ予測又はインター予測のうちの一方）をプロセス２００Ｂの現在の反復のために選択することができる。例えば、符号器は、レート－歪み最適化技法を遂行することができる。本技法では、符号器は、候補予測モード下での候補予測モードのビットレート及び再構成参照ピクチャの歪みに依存する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。選択された予測モードに応じて、符号器は、対応する予測ＢＰＵ２０８及び予測データ２０６を生成することができる。

【００５０】

［0071］プロセス２００Ｂの再構成経路内において、イントラ予測モードが順方向経路内で選択された場合には、予測基準２２４（例えば、現在のピクチャにおいて符号化され、再構成された現在のＢＰＵ）を生成した後に、符号器は、予測基準２２４を後の使用のために（例えば、現在のピクチャの次のＢＰＵの外挿のために）空間的予測段階２０４２に直接供給することができる。インター予測モードが順方向経路内で選択された場合には

、予測基準２２４（例えば、全てのＢＰＵが符号化され、再構成された現在のピクチャ）を生成した後に、符号器は、予測基準２２４をループフィルタ段階２３２に供給することができ、そこで、符号器は、ループフィルタを予測基準２２４に適用し、インター予測によって導入された歪み（例えば、ブロッキングアーチファクト）を低減又は解消することができる。符号器は、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、又は同様のものなどの、様々なループフィルタ技法をループフィルタ段階２３２において適用することができる。ループフィルタリングされた参照ピクチャは、後の使用のために（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基準ピクチャとして用いられるために）バッファ２３４（又は「復号化ピクチャバッファ」）内に記憶され得る。符号器は、１つ以上の参照ピクチャを、時間的予測段階２０４４において用いられるためにバッファ２３４内に記憶することができる。実施形態によっては、符号器は、ループフィルタのパラメータ（例えば、ループフィルタ強度）を、量子化変換係数２１６、予測データ２０６、及び他の情報と共に、２値符号化段階２２６において符号化することができる。

【００５１】

［0072］ 図３Ａは、本開示の実施形態に従う、例示的な復号化プロセス３００Ａの概略図を示す。プロセス３００Ａは、図２Ａにおける圧縮プロセス２００Ａに対応する復元プロセスであることができる。実施形態によっては、プロセス３００Ａはプロセス２００Ａの再構成経路と似たものであることができる。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に復号化することができる。映像ストリーム３０４は映像シーケンス２０２とよく似たものであることができる。しかし、圧縮及び復元プロセス（例えば、図２Ａ～図２Ｂにおける量子化段階２１４）における情報損失のゆえに、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におけるプロセス２００Ａ及び２００Ｂと同様に、復号器は、映像ビットストリーム２２８内に符号化されたピクチャごとに基本処理ユニット（ＢＰＵ）のレベルでプロセス３００Ａを遂行することができる。例えば、復号器はプロセス３００Ａを反復的な仕方で遂行することができ、その場合、復号器は基本処理ユニットをプロセス３００Ａの１回の反復において復号化することができる。実施形態によっては、復号器は、プロセス３００Ａを、映像ビットストリーム２２８内に符号化された各ピクチャの領域（例えば、領域１１４～１１８）のために並行して遂行することができる。

【００５２】

［0073］ 図３Ａにおいて、復号器は、符号化ピクチャの基本処理ユニット（「符号化ＢＰＵ」と称される）に関連付けられた映像ビットストリーム２２８の部分を２値復号化段階３０２に供給することができる。２値復号化段階３０２において、復号器は、当該部分を予測データ２０６及び量子化変換係数２１６に復号化することができる。復号器は、量子化変換係数２１６を逆量子化段階２１８及び逆変換段階２２０に供給し、再構成残差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に供給し、予測ＢＰＵ２０８を生成することができる。復号器は、再構成残差ＢＰＵ２２２を予測ＢＰＵ２０８に加算し、予測基準２２４を生成することができる。実施形態によっては、予測基準２２４をバッファ（例えば、コンピュータメモリ内の復号化ピクチャバッファ）内に記憶することができる。復号器は、予測演算をプロセス３００Ａの次の反復において遂行するために、予測基準２２４を予測段階２０４に供給することができる。

【００５３】

［0074］ 復号器は、符号化ピクチャの各符号化ＢＰＵを復号化し、符号化ピクチャの次の符号化ＢＰＵを符号化するための予測基準２２４を生成するために、プロセス３００Ａを反復的に遂行することができる。符号化ピクチャの全ての符号化ＢＰＵを復号化した後に、復号器は、ピクチャを表示のために映像ストリーム３０４に出力し、映像ビットストリーム２２８内の次の符号化ピクチャを復号化するために進むことができる。

【００５４】

［0075］ ２値復号化段階３０２において、復号器は、符号器によって用いられた２値符

号化技法（例えば、エントロピー符号化、可変長符号化、算術符号化、ハフマン符号化、コンテキスト適応２値算術符号化、又は任意の他の可逆圧縮アルゴリズム）の逆演算を遂行することができる。実施形態によっては、予測データ２０６及び量子化変換係数２１６のほかに、復号器は、例えば、予測モード、予測演算のパラメータ、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）、又は同様のものなどの、他の情報を２値復号化段階３０２において復号化することができる。実施形態によっては、映像ビットストリーム２２８がネットワークを通じてパケットの形で伝送される場合には、復号器は、映像ビットストリーム２２８を、それを２値復号化段階３０２に供給する前にデパケット化することができる。

【００５５】

［0076］ 図３Ｂは、本開示の実施形態に従う、別の例示的な復号化プロセス３００Ｂの概略図を示す。プロセス３００Ｂはプロセス３００Ａから変更され得る。例えば、プロセス３００Ｂは、ハイブリッド映像符号化規格（例えば、Ｈ．２６ｘシリーズ）に準拠した復号器によって用いられ得る。プロセス３００Ａと比べて、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に追加的に分割し、ループフィルタ段階２３２及びバッファ２３４を追加的に含む。

【００５６】

［0077］ プロセス３００Ｂにおいて、復号化中の符号化ピクチャ（「現在のピクチャ」と称される）の符号化基本処理ユニット（「現在のＢＰＵ」と称される）のために、復号器によって２値復号化段階３０２から復号化された予測データ２０６は、いかなる予測モードが符号器によって現在のＢＰＵを符号化するために用いられたのかに依存して、様々な種類のデータを含むことができる。例えば、イントラ予測が符号器によって、現在のＢＰＵを符号化するために用いられた場合には、予測データ２０６は、イントラ予測、イントラ予測演算のパラメータ、又は同様のものを指示する予測モードインジケータ（例えば、フラグ値）を含むことができる。イントラ予測演算のパラメータは、例えば、参照として用いられる１つ以上の隣接ＢＰＵの場所（例えば、座標）、隣接ＢＰＵのサイズ、外挿のパラメータ、原ＢＰＵに対する隣接ＢＰＵの方向、又は同様のものを含むことができる。別の例として、インター予測が符号器によって、現在のＢＰＵを符号化するために用いられた場合には、予測データ２０６は、インター予測、インター予測演算のパラメータ、又は同様のものを指示する予測モードインジケータ（例えば、フラグ値）を含むことができる。インター予測演算のパラメータは、例えば、現在のＢＰＵに関連付けられた参照ピクチャの数、参照ピクチャにそれぞれ関連付けられた重み、それぞれの参照ピクチャ内の１つ以上のマッチング領域の場所（例えば、座標）、マッチング領域にそれぞれ関連付けられた１つ以上の動きベクトル、又は同様のものを含むことができる。

【００５７】

［0078］ 予測モードインジケータに基づいて、復号器は、空間的予測段階２０４２において空間的予測（例えば、イントラ予測）を遂行するべきか、又は時間的予測段階２０４４において時間的予測（例えば、インター予測）を遂行するべきかを決定することができる。このような空間的予測又は時間的予測を遂行することの詳細は図２Ｂにおいて説明されており、以下、繰り返されない。このような空間的予測又は時間的予測を遂行した後に、復号器は予測ＢＰＵ２０８を生成することができる。復号器は、図３Ａにおいて説明されたように、予測ＢＰＵ２０８及び再構成残差ＢＰＵ２２２を加算し、予測基準２２４を生成することができる。

【００５８】

［0079］ プロセス３００Ｂにおいて、復号器は、予測演算をプロセス３００Ｂの次の反復において遂行するために、予測基準２２４を空間的予測段階２０４２又は時間的予測段階２０４４に供給することができる。例えば、現在のＢＰＵが空間的予測段階２０４２においてイントラ予測を用いて復号化される場合には、予測基準２２４（例えば、復号化された現在のＢＰＵ）を生成した後に、復号器は、予測基準２２４を後の使用のために（例

えば、現在のピクチャの次のＢＰＵの外挿のために）空間的予測段階２０４２に直接供給することができる。現在のＢＰＵが時間的予測段階２０４４においてインター予測を用いて復号化される場合には、予測基準２２４（例えば、全てのＢＰＵが復号化された参照ピクチャ）を生成した後に、符号器は、予測基準２２４をループフィルタ段階２３２に供給し、歪み（例えば、ブロッキングアーチファクト）を低減又は解消することができる。復号器は、図２Ｂにおいて説明されたとおりの仕方でループフィルタを予測基準２２４に適用することができる。ループフィルタリングされた参照ピクチャは、後の使用のために（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基準ピクチャとして用いられるために）バッファ２３４（例えば、コンピュータメモリ内の復号化ピクチャバッファ）内に記憶され得る。復号器は、１つ以上の参照ピクチャを、時間的予測段階２０４４において用いられるためにバッファ２３４内に記憶することができる。実施形態によっては、予測データ２０６の予測モードインジケータが、インター予測が現在のＢＰＵを符号化するために用いられたことを指示するときには、予測データはループフィルタのパラメータ（例えば、ループフィルタ強度）をさらに含むことができる。

【００５９】

［0080］図４は、本開示の実施形態に従う、映像を符号化又は復号化するための例示的な装置４００のブロック図である。図４に示されるように、装置４００はプロセッサ４０２を含むことができる。プロセッサ４０２が、本明細書において説明される命令を実行したとき、装置４００は映像符号化又は復号化のための特殊機械になることができる。プロセッサ４０２は、情報を操作又は処理する能力を有する任意の種類の回路機構であることができる。例えば、プロセッサ４０２は、中央処理装置（又は「ＣＰＵ（central processing unit）」）、グラフィック処理装置（又は「ＧＰＵ（graphics processing unit）」）、ニューラル処理装置（「ＮＰＵ（neural processing unit）」）、マイクロコントローラユニット（「ＭＣＵ（microcontroller unit）」）、光プロセッサ、プログラマブル論理コントローラ、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（intellectual property））コア、プログラマブル論理アレイ（ＰＬＡ（Programmable Logic Array））、プログラマブルアレイ論理（ＰＡＬ（Programmable Array Logic））、ジェネリックアレイ論理（ＧＡＬ（Generic Array Logic））、複合プログラマブル論理装置（ＣＰＬＤ（Complex Programmable Logic Device））、フィールドプログラマブルゲートアレイ（ＦＰＧＡ（Field-Programmable Gate Array））、システムオンチップ（ＳｏＣ（System On Chip））、特定用途向け集積回路（ＡＳＩＣ（Application-Specific Integrated Circuit））、又は同様のもののうちの任意の数の任意の組み合わせを含むことができる。実施形態によっては、プロセッサ４０２はまた、単一の論理構成要素としてグループ化されたプロセッサのセットであることもできる。例えば、図４に示されるように、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むことができる。

【００６０】

［0081］装置４００はまた、データ（例えば、命令のセット、コンピュータコード、中間データ、又は同様のもの）を記憶するように構成されたメモリ４０４を含むことができる。例えば、図４に示されるように、記憶されるデータは、プログラム命令（例えば、プロセス２００Ａ、２００Ｂ、３００Ａ、又は３００Ｂにおける段階を実施するためのプログラム命令）、並びに処理のためのデータ（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）を含むことができる。プロセッサ４０２は、（例えば、バス４１０を介して）プログラム命令、及び処理のためのデータにアクセスし、プログラム命令を実行し、処理のためのデータに対する演算又は操作を遂行することができる。メモリ４０４は、高速ランダムアクセス記憶デバイス又は不揮発性記憶デバイスを含むことができる。実施形態によっては、メモリ４０４は、ランダムアクセスメモリ（ＲＡＭ（random-access memory））、リードオンリーメモリ（ＲＯＭ（read-only memory））、光ディスク、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（security digital））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（compact flash））カード、又は同様のもののうちの任意の数の任意の組み合わせを含むことができる。メモリ４０４はまた、単一の論理構成要素としてグループ化されたメモリのグループ（図４には示されていない）であることもできる。

【００６１】

［0082］ バス４１０は、内部バス（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、ペリフェラルコンポーネントインターコネクトエクスプレスポート）、又は同様のものなどの、装置４００の内部の構成要素の間でデータを転送する通信デバイスであることができる。

【００６２】

［0083］ 曖昧さを生じさせることなく説明を容易にするために、プロセッサ４０２及び他のデータ処理回路は本開示においてまとめて「データ処理回路」と称される。データ処理回路は、完全にハードウェアとして、或いはソフトウェア、ハードウェア、又はファームウェアの組み合わせとして実施され得る。加えて、データ処理回路は、単一の独立モジュールであることができるか、或いは装置４００の任意の他の構成要素に完全に、又は部分的に組み合わせられ得る。

【００６３】

［0084］ 装置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、移動通信ネットワーク、又は同様のもの）との有線又は無線通信を提供するためのネットワークインターフェース４０６をさらに含むことができる。実施形態によっては、ネットワークインターフェース４０６は、ネットワークインターフェースコントローラ（ＮＩＣ（network interface controller））、無線周波数（ＲＦ（radio frequency））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（登録商標）アダプタ、赤外線アダプタ、近距離無線通信（「ＮＦＣ（near-field communication）」）アダプタ、セルラーネットワークチップ、又は同様のもののうちの任意の数の任意の組み合わせを含むことができる。

【００６４】

［0085］ 実施形態によっては、任意選択的に、装置４００は、１つ以上の周辺デバイスへの接続を提供するための周辺インターフェース４０８をさらに含むことができる。図４に示されるように、周辺デバイスは、限定するものではないが、カーソル制御デバイス（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、陰極線管ディスプレイ、液晶ディスプレイ、又は発光ダイオードディスプレイ）、映像入力デバイス（例えば、カメラ、又は映像アーカイブに結合された入力インターフェース）、或いは同様のものを含むことができる。

【００６５】

［0086］ 映像コーデック（例えば、プロセス２００Ａ、２００Ｂ、３００Ａ、又は３００Ｂを遂行するコーデック）は、装置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施され得ることに留意されたい。例えば、プロセス２００Ａ、２００Ｂ、３００Ａ、又は３００Ｂの一部又は全ての段階は、メモリ４０４内にロードされ得るプログラム命令などの、装置４００の１つ以上のソフトウェアモジュールとして実施され得る。別の例として、プロセス２００Ａ、２００Ｂ、３００Ａ、又は３００Ｂの一部又は全ての段階は、特殊データ処理回路（例えば、ＦＰＧＡ、ＡＳＩＣ、ＮＰＵ、又は同様のもの）などの、装置４００の１つ以上のハードウェアモジュールとして実施され得る。

【００６６】

［0087］ 量子化及び逆量子化機能ブロック（例えば、図２Ａ又は図２Ｂの量子化２１４及び逆量子化２１８、図３Ａ又は図３Ｂの逆量子化２１８）では、量子化パラメータ（ＱＰ（quantization parameter））が、予測残差に適用される量子化（及び逆量子化）の

量を決定するために用いられる。ピクチャ又はスライスの符号化のために用いられる初期ＱＰ値は、例えば、ピクチャパラメータセット（ＰＰＳ（Picture Parameter Set））内のｉｎｉｔ＿ｑｐ＿ｍｉｎｕｓ２６構文要素を用いて、及びスライスヘッダ内のｓｌｉｃｅ＿ｑｐ＿ｄｅｌｔａ構文要素を用いて、高レベルでシグナリングされ得る。さらに、ＱＰ値は、量子化グループの細分性で送信されたデルタＱＰ値を用いて、ＣＵごとに局所レベルで適応され得る。

【００６７】

［0088］実施形態によっては、ピクチャは、複数の符号化ツリーユニット（ＣＴＵ）に分割され得る。次いで、ＣＴＵは、２分割及び３分割セグメンテーション構造を用いたネスト型マルチタイプツリーを有する４分木（ＳＰＬＩＴ＿ＱＴ）を用いて１つ以上の符号化ユニット（ＣＵ）にさらに分割される。図５は、本開示のいくつかの実施形態に係る、マルチタイプツリー分割モードの例を示す概略図である。図５に示されるように、マルチタイプツリー構造における分割タイプは、４分木分割（ＳＰＬＩＴ＿ＱＴ）５０１、垂直２分割（ＳＰＬＩＴ＿ＢＴ＿ＶＥＲ）５０２、水平２分割（ＳＰＬＩＴ＿ＢＴ＿ＨＯＲ）５０３、垂直３分割（ＳＰＬＩＴ＿ＴＴ＿ＶＥＲ）５０４、及び水平３分割（ＳＰＬＩＴ＿ＴＴ＿ＨＯＲ）５０５を含み得る。マルチタイプツリーのリーフノードは、符号化ユニット（ＣＵ）と称され、正方形又は長方形のいずれかを有し得る。

【００６８】

［0089］図６は、本開示のいくつかの実施形態に係る、ネスト型マルチタイプツリー符号化ツリー構造を有する４分木における区分分割情報のための例示的シグナリング機構を示す概略図である。図６に示されるように、ＣＴＵは、４分木のルートとして扱われ、４分木構造によって最初に区分される。各４分木リーフノード（それを可能にするのに十分大きいとき）は、マルチタイプツリー構造によってさらに区分される。マルチタイプツリー構造において、ノードがさらに区分されるかどうかを示すために、第１のフラグ（例えば、ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｆｌａｇ）がシグナリングされる。ノードがさらに区分されるとき、第２のフラグ（例えば、ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｖｅｒｔｉｃａｌ＿ｆｌａｇ）が分割方向を示すためにシグナリングされ、次いで、第３のフラグ（例えば、ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｂｉｎａｒｙ＿ｆｌａｇ）は、分割が２分割であるか又は３分割であるかを示すためにシグナリングされる。第２のフラグｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｖｅｒｔｉｃａｌ＿ｆｌａｇ及び第３のフラグｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｂｉｎａｒｙ＿ｆｌａｇの値に基づいて、ＣＵのマルチタイプツリー分割モード（ＭｔｔＳｐｌｉｔＭｏｄｅ）が導出され得る。図７は、本開示のいくつかの実施形態に係る、マルチタイプツリー構文要素に基づく例示的なＭｔｔＳｐｌｉｔＭｏｄｅ導出を示す、例示的な表１を示す。

【００６９】

［0090］垂直パイプラインデータユニット（ＶＰＤＵ）は、ピクチャ内の非オーバラップユニットとして定義される。ハードウェア復号器において、連続するＶＰＤＵは、同時に複数のパイプライン段階によって処理される。ＶＰＤＵサイズは、大抵のパイプライン段階においてバッファサイズに大まかに比例し、したがって、ＶＰＤＵサイズを小さく保つことが重要である。大抵のハードウェア復号器では、ＶＰＤＵサイズは、６４×６４ルマサンプルに設定され得る。しかしながら、実施形態によっては、３分木（ＴＴ）及び２分木（ＢＴ）の区分は、ＶＰＤＵサイズの増加につながり得る。本開示に従って、ＶＰＤＵサイズを６４×６４ルマサンプルとして維持するために、ある標準的な区分制限が適用され得る。図８は、本開示のいくつかの実施形態に係る、許可されないＴＴ及びＢＴ分割の例を示す。図８に示されるように、ＴＴ分割は、１２８に等しい幅若しくは高さのいずれか又は１２８に等しい幅及び高さの両方を有するブロックに対しては許可されない。Ｎ≦６４で１２８×ＮのＣＵ（例えば、１２８に等しい幅及び１２８より小さな高さ）について、水平ＢＴは許可されない。Ｎ≦６４でＮ×１２８のＣＵ（例えば、１２８に等しい高さ及び１２８より小さな幅）について、垂直ＢＴは許可されない。

【００７０】

[0091] ＨＥＶＣにおいて必要に応じて、ツリーノードブロックの一部が下又は右ピクチャ境界を超えるとき、ツリーノードブロックは、あらゆる符号化ＣＵの全てのサンプルがピクチャ境界の内側に位置するまで強制分割される。以下の分割規則が、ＶＶＣドラフト７において適用される。

－ ツリーノードブロックの一部が、下及び右ピクチャ境界の両方を超える場合、

－ ブロックがＱＴノードであり、且つブロックのサイズが最小ＱＴサイズより大きい場合、ブロックは、ＱＴ分割モードで強制分割される。

－ それ以外の場合、ブロックは、ＳＰＬＩＴ＿ＢＴ＿ＨＯＲモードで強制分割される。

－ それ以外の場合で、ツリーノードブロックの一部が下ピクチャ境界を超える場合、

－ ブロックがＱＴノードであり、ブロックのサイズが最小ＱＴサイズより大きく、且つブロックのサイズが最大ＢＴサイズより大きい場合、ブロックは、ＱＴ分割モードで強制分割される。

－ それ以外の場合で、ブロックがＱＴノードであり、ブロックのサイズが最小ＱＴサイズより大きく、且つブロックのサイズが最大ＢＴサイズ以下である場合、ブロックは、ＱＴ分割モード又はＳＰＬＩＴ＿ＢＴ＿ＨＯＲモードで強制分割される。

－ それ以外の（ブロックがＢＴノードであるか又はブロックのサイズが最小ＱＴサイズ以下である）場合、ブロックは、ＳＰＬＩＴ＿ＢＴ＿ＨＯＲモードで強制分割される。

－ それ以外の場合で、ツリーノードブロックの一部が右ピクチャ境界を超える場合、

－ ブロックがＱＴノードであり、ブロックのサイズが最小ＱＴサイズより大きく、且つブロックのサイズが最大ＢＴサイズより大きい場合、ブロックは、ＱＴ分割モードで強制分割される。

－ それ以外の場合で、ブロックがＱＴノードであり、ブロックのサイズが最小ＱＴサイズより大きく、且つブロックのサイズが最大ＢＴサイズ以下である場合、ブロックは、ＱＴ分割モード又はＳＰＬＩＴ＿ＢＴ＿ＶＥＲモードで強制分割される。

－ それ以外の（例えば、ブロックがＢＴノードであるか又はブロックのサイズが最小ＱＴサイズ以下である）場合、ブロックは、ＳＰＬＩＴ＿ＢＴ＿ＶＥＲモードで強制分割される。

【００７１】

[0092] 図９は、本開示のいくつかの実施形態に係る、ピクチャ境界上の例示的ブロック区分を示す。図９に示されるように、ＣＴＵ９１１について、ＳＰＬＩＴ＿ＱＴ又はＳＰＬＩＴ＿ＢＴ＿ＶＥＲのいずれかが実行され得る。ＣＴＵ９１３について、ＳＰＬＩＴ＿ＱＴが許可される場合、ＳＰＬＩＴ＿ＱＴが実行され得る。ＳＰＬＩＴ＿ＱＴが許可されない場合、ＳＰＬＩＴ＿ＢＴ＿ＨＯＲが実行され得る。ＣＴＵ９１５については、ＳＰＬＩＴ＿ＱＴ又はＳＰＬＩＴ＿ＢＴ＿ＨＯＲのいずれかが実行され得る。

【００７２】

[0093] ＶＶＣドラフト７には、ブロック区分に関連する２つの節がある。第１は、６．４節であり、ブロックが４分木、２分木、又は３分木を用いて分割され得るかどうかを定義する。６．４節の出力は、変数ａｌｌｏｗＳｐｌｉｔＱｔ、ａｌｌｏｗＳｐｌｉｔＢｔＨｏｒ、ａｌｌｏｗＳｐｌｉｔＢｔＶｅｒ、ａｌｌｏｗＳｐｌｉｔＴｔＨｏｒ、及びａｌｌｏｗＳｐｌｉｔＴｔＶｅｒである。これらの変数は、７．３．９．４節において図１２の表４に示されるように、対応するＣＵレベル分割フラグ（図１２の表４中のボックス１２０１～１２０４によって示される）がシグナリングされるか否かを決定するために用いられる。

【００７３】

[0094] ＶＶＣドラフト７の６．４．１節は、以下のように記載されている。

６．４ 可用性プロセス

６．４．１ 許可される４分割プロセス

本プロセスへの入力は、以下の通りである。

－ ルマサンプル内の符号化ブロックサイズｃｂＳｉｚｅ

－ マルチタイプツリー深さｍｔｔＤｅｐｔｈ
－ 符号化ツリーノードを区分するためにシングルツリー（ＳＩＮＧＬＥ＿ＴＲＥＥ）が用いられるか、又はデュアルツリーが用いられるか、及びデュアルツリーが用いられるときに、ルマ成分（ＤＵＡＬ＿ＴＲＥＥ＿ＬＵＭＡ）が現在処理されるか又はクロマ成分（ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡ）が現在処理されるかを指定する、変数ｔｒｅｅＴｙｐｅ
－ 符号化ツリーノードの内側の符号化ユニットに対して、イントラ符号化モード（ＭＯＤＥ＿ＩＮＴＲＡ）、ＩＢＣ符号化モード（ＭＯＤＥ＿ＩＢＣ）、及びインター符号化モードが用いられ得るか（ＭＯＤＥ＿ＴＹＰＥ＿ＡＬＬ）、又はイントラ符号化モード及びＩＢＣ符号化モードのみが用いられ得るか（ＭＯＤＥ＿ＴＹＰＥ＿ＩＮＴＲＡ）、又はインター符号化モードのみが用いられ得るか（ＭＯＤＥ＿ＴＹＰＥ＿ＩＮＴＥＲ）を指定する、変数ｍｏｄｅＴｙｐｅ

【００７４】

[0095] 本プロセスの出力は、変数ａｌｌｏｗＳｐｌｉｔＱｔである。変数ａｌｌｏｗＳｐｌｉｔＱｔは、以下のように導出される。
－ 以下の条件のうちの１つ以上が真である場合、ａｌｌｏｗＳｐｌｉｔＱｔは、ＦＡＬＳＥに等しく設定される。
－ ｔｒｅｅＴｙｐｅが、ＳＩＮＧＬＥ＿ＴＲＥＥ又はＤＵＡＬ＿ＴＲＥＥ＿ＬＵＭＡに等しく、且つｃｂＳｉｚｅが、ＭｉｎＱｔＳｉｚｅＹ以下である。
－ ｔｒｅｅＴｙｐｅが、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しく、且つｃｂＳｉｚｅ／ＳｕｂＷｉｄｔｈＣが、ＭｉｎＱｔＳｉｚｅＣ以下である。
－ ｍｔｔＤｅｐｔｈが、０に等しくない。
－ ｔｒｅｅＴｙｐｅが、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しく、且つ（ｃｂＳｉｚｅ／ＳｕｂＷｉｄｔｈＣ）が、４以下である。
－ ｔｒｅｅＴｙｐｅが、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しく、且つｍｏｄｅＴｙｐｅが、ＭＯＤＥ＿ＴＹＰＥ＿ＩＮＴＲＡに等しい。
－ それ以外の場合、ａｌｌｏｗＳｐｌｉｔＱｔは、ＴＲＵＥに等しく設定される。

【００７５】

[0096] ＶＶＣドラフト７の６．４．２節は、以下のように記載されている。

６．４．２ 許可される２分割プロセス

本プロセスへの入力は、以下の通りである。
－ ２分割モードｂｔＳｐｌｉｔ
－ ルマサンプル内の符号化ブロック幅ｃｂＷｉｄｔｈ
－ ルマサンプル内の符号化ブロック高さｃｂＨｅｉｇｈｔ
－ ピクチャの左上ルマサンプルに対する、考えられる符号化ブロックの左上ルマサンプルの位置（ｘ０，ｙ０）
－ マルチタイプツリー深さｍｔｔＤｅｐｔｈ
－ オフセット付き最大マルチタイプツリー深さｍａｘＭｔｔＤｅｐｔｈ
－ 最大２分木サイズｍａｘＢｔＳｉｚｅ
－ 最小４分木サイズｍｉｎＱｔＳｉｚｅ
－ 区分インデックスｐａｒｔＩｄｘ
－ 符号化ツリーノードを区分するためにシングルツリー（ＳＩＮＧＬＥ＿ＴＲＥＥ）が用いられるか、又はデュアルツリーが用いられるか、及びデュアルツリーが用いられるときに、ルマ成分（ＤＵＡＬ＿ＴＲＥＥ＿ＬＵＭＡ）が現在処理されるか又はクロマ成分（ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡ）が現在処理されるかを指定する、変数ｔｒｅｅＴｙｐｅ
－ 符号化ツリーノードの内側の符号化ユニットに対して、イントラ符号化モード（ＭＯＤＥ＿ＩＮＴＲＡ）、ＩＢＣ符号化モード（ＭＯＤＥ＿ＩＢＣ）、及びインター符号化モードが用いられ得るか（ＭＯＤＥ＿ＴＹＰＥ＿ＡＬＬ）、又はイントラ符号化モード及びＩＢＣ符号化モードのみが用いられ得るか（ＭＯＤＥ＿ＴＹＰＥ＿ＩＮＴＲＡ）、又は

インター符号化モードのみが用いられ得るか（ＭＯＤＥ＿ＴＹＰＥ＿ＩＮＴＥＲ）を指定する、変数ｍｏｄｅＴｙｐｅ
【００７６】
［0097］本プロセスの出力は、変数ａｌｌｏｗＢｔＳｐｌｉｔである。図１０は、本開示のいくつかの実施形態に係る、ｂｔＳｐｌｉｔに基づく変数ｐａｒａｌｌｅｌＴｔＳｐｌｉｔ及びｃｂＳｉｚｅの例示的仕様を示す例示的な表２を示す。
【００７７】
［0098］変数ａｌｌｏｗＢｔＳｐｌｉｔは、以下のように導出される。
－ 以下の条件のうちの１つ以上が真である場合、ａｌｌｏｗＢｔＳｐｌｉｔは、ＦＡＬＳＥに等しく設定される。
－ ｃｂＳｉｚｅが、ＭｉｎＢｔＳｉｚｅＹ以下である。
－ ｃｂＷｉｄｔｈが、ｍａｘＢｔＳｉｚｅより大きい。
－ ｃｂＨｅｉｇｈｔが、ｍａｘＢｔＳｉｚｅより大きい。
－ ｍｔｔＤｅｐｔｈが、ｍａｘＭｔｔＤｅｐｔｈ以上である。
－ ｔｒｅｅＴｙｐｅが、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しく、且つ（ｃｂＷｉｄｔｈ／ＳｕｂＷｉｄｔｈＣ）×（ｃｂＨｅｉｇｈｔ／ＳｕｂＨｅｉｇｈｔＣ）が、１６以下である。
－ ｔｒｅｅＴｙｐｅが、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しく、（ｃｂＷｉｄｔｈ／ＳｕｂＷｉｄｔｈＣ）が、４に等しく、且つｂｔＳｐｌｉｔが、ＳＰＬＩＴ＿ＢＴ＿ＶＥＲに等しい。
－ ｔｒｅｅＴｙｐｅが、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しく、且つｍｏｄｅＴｙｐｅが、ＭＯＤＥ＿ＴＹＰＥ＿ＩＮＴＲＡに等しい。
－ ｃｂＷｉｄｔｈ×ｃｂＨｅｉｇｈｔが、３２に等しく、且つｍｏｄｅＴｙｐｅが、ＭＯＤＥ＿ＴＹＰＥ＿ＩＮＴＥＲに等しい。
－ それ以外の場合で、以下の条件の全てが真である場合、ａｌｌｏｗＢｔＳｐｌｉｔは、ＦＡＬＳＥに等しく設定される。
－ ｂｔＳｐｌｉｔが、ＳＰＬＩＴ＿ＢＴ＿ＶＥＲに等しい。
－ ｙ０＋ｃｂＨｅｉｇｈｔが、ｐｉｃ＿ｈｅｉｇｈｔ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きい。
－ それ以外の場合で、以下の条件の全てが真である場合、ａｌｌｏｗＢｔＳｐｌｉｔは、ＦＡＬＳＥに等しく設定される。
－ ｂｔＳｐｌｉｔが、ＳＰＬＩＴ＿ＢＴ＿ＶＥＲに等しい。
－ ｃｂＨｅｉｇｈｔが、６４より大きい。
－ ｘ０＋ｃｂＷｉｄｔｈが、ｐｉｃ＿ｗｉｄｔｈ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きい。
－ それ以外の場合で、以下の条件の全てが真である場合、ａｌｌｏｗＢｔＳｐｌｉｔは、ＦＡＬＳＥに等しく設定される。
－ ｂｔＳｐｌｉｔが、ＳＰＬＩＴ＿ＢＴ＿ＨＯＲに等しい。
－ ｃｂＷｉｄｔｈが、６４より大きい。
－ ｙ０＋ｃｂＨｅｉｇｈｔが、ｐｉｃ＿ｈｅｉｇｈｔ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きい。
－ それ以外の場合で、以下の条件の全てが真である場合、ａｌｌｏｗＢｔＳｐｌｉｔは、ＦＡＬＳＥに等しく設定される。
－ ｘ０＋ｃｂＷｉｄｔｈが、ｐｉｃ＿ｗｉｄｔｈ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きい。
－ ｙ０＋ｃｂＨｅｉｇｈｔが、ｐｉｃ＿ｈｅｉｇｈｔ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きい。
－ ｃｂＷｉｄｔｈが、ｍｉｎＱｔＳｉｚｅより大きい。
－ それ以外の場合で、以下の条件の全てが真である場合、ａｌｌｏｗＢｔＳｐｌｉｔは、ＦＡＬＳＥに等しく設定される。

－ ｂｔＳｐｌｉｔが、ＳＰＬＩＴ＿ＢＴ＿ＨＯＲに等しい。
－ ｘ０＋ｃｂＷｉｄｔｈが、ｐｉｃ＿ｗｉｄｔｈ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きい。
－ ｙ０＋ｃｂＨｅｉｇｈｔが、ｐｉｃ＿ｈｅｉｇｈｔ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓ以下である。
－ それ以外の場合で、以下の条件の全てが真である場合、ａｌｌｏｗＢｔＳｐｌｉｔは、ＦＡＬＳＥに等しく設定される。
－ ｍｔｔＤｅｐｔｈが、０より大きい。
－ ｐａｒｔＩｄｘは、１に等しい。
－ ＭｔｔＳｐｌｉｔＭｏｄｅ［ｘ０］［ｙ０］［ｍｔｔＤｅｐｔｈ－１］が、ｐａｒａｌｌｅｌＴｔＳｐｌｉｔに等しい。
－ それ以外の場合で、以下の条件の全てが真である場合、ａｌｌｏｗＢｔＳｐｌｉｔは、ＦＡＬＳＥに等しく設定される。
－ ｂｔＳｐｌｉｔが、ＳＰＬＩＴ＿ＢＴ＿ＶＥＲに等しい。
－ ｃｂＷｉｄｔｈが、６４以下である。
－ ｃｂＨｅｉｇｈｔが、６４より大きい。
－ それ以外の場合で、以下の条件の全てが真である場合、ａｌｌｏｗＢｔＳｐｌｉｔは、ＦＡＬＳＥに等しく設定される。
－ ｂｔＳｐｌｉｔが、ＳＰＬＩＴ＿ＢＴ＿ＨＯＲに等しい。
－ ｃｂＷｉｄｔｈが、６４より大きい。
－ ｃｂＨｅｉｇｈｔが、６４以下である。
－ それ以外の場合、ａｌｌｏｗＢｔＳｐｌｉｔは、ＴＲＵＥに等しく設定される。

【００７８】

［0099］ ＶＶＣドラフト７の６．４．３節は、以下のように記載される。

６．４．３ 許可される３分割プロセス

本プロセスへの入力は、以下の通りである。
－ ３分割モードｔｔＳｐｌｉｔ
－ ルマサンプル内の符号化ブロック幅ｃｂＷｉｄｔｈ
－ ルマサンプル内の符号化ブロック高さｃｂＨｅｉｇｈｔ
－ ピクチャの左上ルマサンプルに対する、考えられる符号化ブロックの左上ルマサンプルの位置（ｘ０，ｙ０）
－ マルチタイプツリー深さｍｔｔＤｅｐｔｈ
－ オフセット付き最大マルチタイプツリー深さｍａｘＭｔｔＤｅｐｔｈ
－ 最大３分木サイズｍａｘＴｔＳｉｚｅ
－ 符号化ツリーノードを区分するためにシングルツリー（ＳＩＮＧＬＥ＿ＴＲＥＥ）が用いられるか、又はデュアルツリーが用いられるか、及びデュアルツリーが用いられるときに、ルマ成分（ＤＵＡＬ＿ＴＲＥＥ＿ＬＵＭＡ）が現在処理されるか又はクロマ成分（ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡ）が現在処理されるかを指定する、変数ｔｒｅｅＴｙｐｅ
－ 符号化ツリーノードの内側の符号化ユニットに対して、イントラ符号化モード（ＭＯＤＥ＿ＩＮＴＲＡ）、ＩＢＣ符号化モード（ＭＯＤＥ＿ＩＢＣ）、及びインター符号化モードが用いられ得るか（ＭＯＤＥ＿ＴＹＰＥ＿ＡＬＬ）、又はイントラ符号化モード及びＩＢＣ符号化モードのみが用いられ得るか（ＭＯＤＥ＿ＴＹＰＥ＿ＩＮＴＲＡ）、又はインター符号化モードのみが用いられ得るか（ＭＯＤＥ＿ＴＹＰＥ＿ＩＮＴＥＲ）を指定する、変数ｍｏｄｅＴｙｐｅ

【００７９】

［00100］ 本プロセスの出力は、変数ａｌｌｏｗＴｔＳｐｌｉｔである。図１１は、本開示のいくつかの実施形態に係る、ｔｔＳｐｌｉｔに基づく変数ｃｂＳｉｚｅの例示的仕様を示す例示的な表３を示す。

【００８０】

[00101] 変数ａｌｌｏｗＴｔＳｐｌｉｔは、以下のように導出される。
－ 以下の条件のうちの１つ以上が真である場合、ａｌｌｏｗＴｔＳｐｌｉｔは、ＦＡＬＳＥに等しく設定される。
－ ｃｂＳｉｚｅが、２×ＭｉｎＴｔＳｉｚｅＹ以下である。
－ ｃｂＷｉｄｔｈが、Ｍｉｎ（６４，ｍａｘＴｔＳｉｚｅ）より大きい。
－ ｃｂＨｅｉｇｈｔが、Ｍｉｎ（６４，ｍａｘＴｔＳｉｚｅ）より大きい。
－ ｍｔｔＤｅｐｔｈが、ｍａｘＭｔｔＤｅｐｔｈ以上である。
－ ｘ０＋ｃｂＷｉｄｔｈが、ｐｉｃ＿ｗｉｄｔｈ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きい。
－ ｙ０＋ｃｂＨｅｉｇｈｔが、ｐｉｃ＿ｈｅｉｇｈｔ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きい。
－ ｔｒｅｅＴｙｐｅが、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しく、且つ（ｃｂＷｉｄｔｈ／ＳｕｂＷｉｄｔｈＣ）×（ｃｂＨｅｉｇｈｔ／ＳｕｂＨｅｉｇｈｔＣ）が、３２以下である。
－ ｔｒｅｅＴｙｐｅが、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しく、（ｃｂＷｉｄｔｈ／ＳｕｂＷｉｄｔｈＣ）が、８に等しく、且つｔｔＳｐｌｉｔが、ＳＰＬＩＴ＿ＴＴ＿ＶＥＲに等しい。
－ ｔｒｅｅＴｙｐｅが、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しく、且つｍｏｄｅＴｙｐｅが、ＭＯＤＥ＿ＴＹＰＥ＿ＩＮＴＲＡに等しい。
－ ｃｂＷｉｄｔｈ×ｃｂＨｅｉｇｈｔが、６４に等しく、且つｍｏｄｅＴｙｐｅが、ＭＯＤＥ＿ＴＹＰＥ＿ＩＮＴＥＲに等しい。
－ それ以外の場合、ａｌｌｏｗＴｔＳｐｌｉｔは、ＴＲＵＥに等しく設定される。

【００８１】

[00102] 図１２は、本開示のいくつかの実施形態に係る、ＶＶＣドラフト７の例示的な７．３．９．４節の符号化ツリー構文（斜体及び影を加えて強調）を示す例示的な表４を示す。

【００８２】

[00103] 変数ａｌｌｏｗＳｐｌｉｔＱｔ、ａｌｌｏｗＳｐｌｉｔＢｔＶｅｒ、ａｌｌｏｗＳｐｌｉｔＢｔＨｏｒ、ａｌｌｏｗＳｐｌｉｔＴｔＶｅｒ、及びａｌｌｏｗＳｐｌｉｔＴｔＨｏｒは、以下のように導出される。
－ 条項６．４．１に指定されるように許可される４分割プロセスは、ｃｂＷｉｄｔｈに等しく設定された符号化ブロックサイズｃｂＳｉｚｅ、現在のマルチタイプツリー深さｍｔｔＤｅｐｔｈ、ｔｒｅｅＴｙｐｅＣｕｒｒ、及びｍｏｄｅＴｙｐｅＣｕｒｒを入力として用いて呼び出され、出力は、ａｌｌｏｗＳｐｌｉｔＱｔに割り当てる。
－ 変数ｍｉｎＱｔＳｉｚｅ、ｍａｘＢｔＳｉｚｅ、ｍａｘＴｔＳｉｚｅ、及びｍａｘＭｔｔＤｅｐｔｈは、以下のように導出される。
－ ｔｒｅｅＴｙｐｅがＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しい場合、ｍｉｎＱｔＳｉｚｅ、ｍａｘＢｔＳｉｚｅ、ｍａｘＴｔＳｉｚｅ、及びｍａｘＭｔｔＤｅｐｔｈは、それぞれＭｉｎＱｔＳｉｚｅＣ、ＭａｘＢｔＳｉｚｅＣ、ＭａｘＴｔＳｉｚｅＣ、及びＭａｘＭｔｔＤｅｐｔｈＣ＋ｄｅｐｔｈＯｆｆｓｅｔに等しく設定される。
－ それ以外の場合、ｍｉｎＱｔＳｉｚｅ、ｍａｘＢｔＳｉｚｅ、ｍａｘＴｔＳｉｚｅ、及びｍａｘＭｔｔＤｅｐｔｈは、それぞれＭｉｎＱｔＳｉｚｅＹ、ＭａｘＢｔＳｉｚｅＹ、ＭａｘＴｔＳｉｚｅＹ、及びＭａｘＭｔｔＤｅｐｔｈＹ＋ｄｅｐｔｈＯｆｆｓｅｔに等しく設定される。
－ 条項６．４．２に指定されるように許可される２分割プロセスは、２分割モードＳＰＬＩＴ＿ＢＴ＿ＶＥＲ、符号化ブロック幅ｃｂＷｉｄｔｈ、符号化ブロック高さｃｂＨｅｉｇｈｔ、位置（ｘ０，ｙ０）、現在のマルチタイプツリー深さｍｔｔＤｅｐｔｈ、オフセット付き最大マルチタイプツリー深さｍａｘＭｔｔＤｅｐｔｈ、最大２分木サイズｍａｘＢｔＳｉｚｅ、最小４分木サイズｍｉｎＱｔＳｉｚｅ、現在の区分インデックスｐａｒｔＩｄｘ、ｔｒｅｅＴｙｐｅＣｕｒｒ、及びｍｏｄｅＴｙｐｅＣｕｒｒを入力として用

いて呼び出され、出力がａｌｌｏｗＳｐｌｉｔＢｔＶｅｒに割り当てられる。
－　条項６．４．２に指定されるように許可される２分割プロセスは、２分割モードＳＰＬＩＴ＿ＢＴ＿ＨＯＲ、符号化ブロック高さｃｂＨｅｉｇｈｔ、符号化ブロック幅ｃｂＷｉｄｔｈ、位置（ｘ０，ｙ０）、現在のマルチタイプツリー深さｍｔｔＤｅｐｔｈ、オフセット付き最大マルチタイプツリー深さｍａｘＭｔｔＤｅｐｔｈ、最大２分木サイズｍａｘＢｔＳｉｚｅ、最小４分木サイズｍｉｎＱｔＳｉｚｅ、現在の区分インデックスｐａｒｔＩｄｘ、ｔｒｅｅＴｙｐｅＣｕｒｒ、及びｍｏｄｅＴｙｐｅＣｕｒｒを入力として用いて呼び出され、出力がａｌｌｏｗＳｐｌｉｔＢｔＨｏｒに割り当てられる。
－　条項６．４．３に指定されるように許可される３分割プロセスは、３分割モードＳＰＬＩＴ＿ＴＴ＿ＶＥＲ、符号化ブロック幅ｃｂＷｉｄｔｈ、符号化ブロック高さｃｂＨｅｉｇｈｔ、位置（ｘ０，ｙ０）、現在のマルチタイプツリー深さｍｔｔＤｅｐｔｈ、オフセット付き最大マルチタイプツリー深さｍａｘＭｔｔＤｅｐｔｈ、最大３分木サイズｍａｘＴｔＳｉｚｅ、ｔｒｅｅＴｙｐｅＣｕｒｒ、及びｍｏｄｅＴｙｐｅＣｕｒｒを入力として用いて呼び出され、出力がａｌｌｏｗＳｐｌｉｔＴｔＶｅｒに割り当てられる。
－　条項６．４．３に指定されるように許可される３分割プロセスは、３分割モードＳＰＬＩＴ＿ＴＴ＿ＨＯＲ、符号化ブロック高さｃｂＨｅｉｇｈｔ、符号化ブロック幅ｃｂＷｉｄｔｈ、位置（ｘ０，ｙ０）、現在のマルチタイプツリー深さｍｔｔＤｅｐｔｈ、オフセット付き最大マルチタイプツリー深さｍａｘＭｔｔＤｅｐｔｈ、最大３分木サイズｍａｘＴｔＳｉｚｅ、ｔｒｅｅＴｙｐｅＣｕｒｒ、及びｍｏｄｅＴｙｐｅＣｕｒｒを入力として用いて呼び出され、出力がａｌｌｏｗＳｐｌｉｔＴｔＨｏｒに割り当てられる。
【００８３】
[00104] ０に等しい構文要素ｓｐｌｉｔ＿ｃｕ＿ｆｌａｇは、符号化ユニットは分割されないことを指定する。１に等しい構文要素ｓｐｌｉｔ＿ｃｕ＿ｆｌａｇは、符号化ユニットが、構文要素ｓｐｌｉｔ＿ｑｔ＿ｆｌａｇによって示されるように４分割を用いて４つの符号化ユニットに、又は構文要素ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｂｉｎａｒｙ＿ｆｌａｇによって示されるように２分割を用いて２つの符号化ユニットに、若しくは３分割を用いて３つの符号化ユニットに分割されることを指定する。２分割又は３分割は、構文要素ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｖｅｒｔｉｃａｌ＿ｆｌａｇによって示されるように、垂直又は水平のいずれかであり得る。
【００８４】
[00105] 構文要素ｓｐｌｉｔ＿ｃｕ＿ｆｌａｇが存在しないとき、ｓｐｌｉｔ＿ｃｕ＿ｆｌａｇの値は、以下のように推定される。
－　以下の条件のうちの１つ以上が真である場合、ｓｐｌｉｔ＿ｃｕ＿ｆｌａｇの値は、１に等しいと推定される。
　－　ｘ０＋ｃｂＷｉｄｔｈが、ｐｉｃ＿ｗｉｄｔｈ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きい。
　－　ｙ０＋ｃｂＨｅｉｇｈｔが、ｐｉｃ＿ｈｅｉｇｈｔ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きい。
－　それ以外の場合、ｓｐｌｉｔ＿ｃｕ＿ｆｌａｇの値は、０に等しいと推定される。
【００８５】
[00106] 構文要素ｓｐｌｉｔ＿ｑｔ＿ｆｌａｇは、符号化ユニットが半分の水平及び垂直サイズを有する符号化ユニットに分割されるかどうかを指定する。
【００８６】
[00107] 構文要素ｓｐｌｉｔ＿ｑｔ＿ｆｌａｇが存在しないとき、以下を適用する。
－　ａｌｌｏｗＳｐｌｉｔＱｔがＴＲＵＥに等しい場合、ｓｐｌｉｔ＿ｑｔ＿ｆｌａｇの値は、１に等しいと推定される。
－　それ以外の場合、ｓｐｌｉｔ＿ｑｔ＿ｆｌａｇの値は、０に等しいと推定される。
【００８７】
[00108] ０に等しい構文要素ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｖｅｒｔｉｃａｌ＿ｆｌａｇは、符号化ユニットが水平方向に分割されることを指定する。１に等しい構文要素ｍｔ

ｔ＿ｓｐｌｉｔ＿ｃｕ＿ｖｅｒｔｉｃａｌ＿ｆｌａｇは、符号化ユニットが垂直方向に分割されることを指定する。

【００８８】

[00109] 構文要素ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｖｅｒｔｉｃａｌ＿ｆｌａｇが存在しないとき、以下のように推定される。

－ ａｌｌｏｗＳｐｌｉｔＢｔＨｏｒがＴＲＵＥに等しいか又はａｌｌｏｗＳｐｌｉｔＴｔＨｏｒがＴＲＵＥに等しい場合、ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｖｅｒｔｉｃａｌ＿ｆｌａｇの値は、０に等しいと推定される。

－ それ以外の場合、ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｖｅｒｔｉｃａｌ＿ｆｌａｇの値は、１に等しいと推定される。

【００８９】

[00110] ０に等しい構文要素ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｂｉｎａｒｙ＿ｆｌａｇは、符号化ユニットが３分割を用いて３つの符号化ユニットに分割されることを指定する。１に等しい構文要素ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｂｉｎａｒｙ＿ｆｌａｇは、符号化ユニットが２分割を用いて２つの符号化ユニットに分割されることを指定する。

【００９０】

[00111] 構文要素ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｂｉｎａｒｙ＿ｆｌａｇが存在しないとき、以下のように推定される。

－ ａｌｌｏｗＳｐｌｉｔＢｔＶｅｒがＦＡＬＳＥに等しく、且つａｌｌｏｗＳｐｌｉｔＢｔＨｏｒがＦＡＬＳＥに等しい場合、ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｂｉｎａｒｙ＿ｆｌａｇの値は、０に等しいと推定される。

－ それ以外の場合に、ａｌｌｏｗＳｐｌｉｔＴｔＶｅｒがＦＡＬＳＥに等しく、且つａｌｌｏｗＳｐｌｉｔＴｔＨｏｒがＦＡＬＳＥに等しい場合、ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｂｉｎａｒｙ＿ｆｌａｇの値は、１に等しいと推定される。

－ それ以外の場合に、ａｌｌｏｗＳｐｌｉｔＢｔＨｏｒがＴＲＵＥに等しく、且つａｌｌｏｗＳｐｌｉｔＴｔＶｅｒがＴＲＵＥに等しい場合、ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｂｉｎａｒｙ＿ｆｌａｇの値は、！ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｖｅｒｔｉｃａｌ＿ｆｌａｇに等しいと推定される。

－ それ以外の（ａｌｌｏｗＳｐｌｉｔＢｔＶｅｒがＴＲＵＥに等しく、且つａｌｌｏｗＳｐｌｉｔＴｔＨｏｒがＴＲＵＥに等しい）場合、ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｂｉｎａｒｙ＿ｆｌａｇの値は、ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｖｅｒｔｉｃａｌ＿ｆｌａｇに等しいと推定される。

【００９１】

[00112] 図１４は、本開示のいくつかの実施形態に係る、ＭｔｔＳｐｌｉｔＭｏｄｅの例示的仕様を示す例示的な表５を示す。変数ＭｔｔＳｐｌｉｔＭｏｄｅ［ｘ］［ｙ］［ｍｔｔＤｅｐｔｈ］は、ｘ＝ｘ０．．ｘ０＋ｃｂＷｉｄｔｈ－１及びｙ＝ｙ０．．ｙ０＋ｃｂＨｅｉｇｈｔ－１について表４に定義されるように、構文要素ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｖｅｒｔｉｃａｌ＿ｆｌａｇの値から、且つ構文要素ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｂｉｎａｒｙ＿ｆｌａｇの値から導出される。

【００９２】

[00113] ＭｔｔＳｐｌｉｔＭｏｄｅ［ｘ０］［ｙ０］［ｍｔｔＤｅｐｔｈ］は、マルチタイプツリー内の符号化ユニットの水平２分割、垂直２分割、水平３分割、及び垂直３分割を表す。アレイインデックスｘ０、ｙ０は、ピクチャの左上ルマサンプルに対する、考えられる符号化ブロックの左上ルマサンプルの位置（ｘ０，ｙ０）を指定する。図１３は、本開示のいくつかの実施形態に係る、ＭｔｔＳｐｌｉｔＭｏｄｅによって示されるマルチタイプツリー分割モードの例を示す。図１３に示されるように、マルチタイプツリー分割モードは、垂直２分割（ＳＰＬＩＴ＿ＢＴ＿ＶＥＲ）１３０１、水平２分割（ＳＰＬＩＴ＿ＢＴ＿ＨＯＲ）１３０２、垂直３分割（ＳＰＬＩＴ＿ＴＴ＿ＶＥＲ）１３０３、及び水平３分割（ＳＰＬＩＴ＿ＴＴ＿ＨＯＲ）１３０４を含み得る。

【００９３】

［00114］４分木分割、２分木分割、及び３分木分割についてのＣＴＵサイズ、最小ブロックサイズ、及びブロックサイズ制限は、シーケンスパラメータセット又はピクチャヘッダのいずれかにおいてシグナリングされることに留意されたい。

【００９４】

［00115］図１５は、本開示のいくつかの実施形態に係る、ＶＶＣドラフト７の例示的な７．３．２．３節のシーケンスパラメータセットＲＢＳＰ構文を示す例示的な表６を示す。図１６は、本開示のいくつかの実施形態に係る、ＶＶＣドラフト７の例示的な７．３．２．６節のピクチャヘッダＲＢＳＰ構文を示す例示的な表７を示す。

【００９５】

［00116］いくつかの実施形態によれば、ツリーノードブロックの一部が下又は右ピクチャ境界を超えるとき、ツリーノードブロックは、あらゆる符号化ブロックの全てのサンプルがピクチャ境界の内側に位置するまで強制分割される。しかしながら、ピクチャ境界に位置し、且つピクチャ境界を超えるサンプルを含むブロックに対して全てのツリー分割モードが許可されない場合がある。図１７は、本開示のいくつかの実施形態に係る、ＱＴ、ＴＴ、又はＢＴ分割のいずれもピクチャ境界で許可されない例示的ブロックを示す。例えば、図１７は、ＱＴ、ＴＴ、又はＢＴ分割のいずれも許可されないピクチャ境界におけるＣＴＵ１７０１、ＣＴＵ１７０３、及びＣＴＵ１７０５を示す。

【００９６】

［00117］全てのツリー分割モードが許可されない第１の例示的な場合として、ＣＴＵサイズ及び最小ＱＴサイズの両方が１２８に設定されるとき、変数ａｌｌｏｗＳｐｌｉｔＱｔ、ａｌｌｏｗＳｐｌｉｔＢｔＨｏｒ、ａｌｌｏｗＳｐｌｉｔＢｔＶｅｒ、ａｌｌｏｗＳｐｌｉｔＴｔＨｏｒ、及びａｌｌｏｗＳｐｌｉｔＴｔＶｅｒの全てが偽に設定される。

【００９７】

［00118］変数ａｌｌｏｗＳｐｌｉｔＱｔは、以下の条件（斜体で強調）に起因して偽に設定される。

６．４．１　許可される４分割プロセス

．．．

－　以下の条件のうちの１つ以上が真である場合、ａｌｌｏｗＳｐｌｉｔＱｔは、ＦＡＬＳＥに等しく設定される。

－　ｔｒｅｅＴｙｐｅが、ＳＩＮＧＬＥ＿ＴＲＥＥ又はＤＵＡＬ＿ＴＲＥＥ＿ＬＵＭＡに等しく、且つｃｂＳｉｚｅが、ＭｉｎＱｔＳｉｚｅＹ以下である。

－　ｔｒｅｅＴｙｐｅが、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しく、且つｃｂＳｉｚｅ／ＳｕｂＷｉｄｔｈＣが、ＭｉｎＱｔＳｉｚｅＣ以下である。

．．．

【００９８】

［00119］変数ａｌｌｏｗＳｐｌｉｔＢｔＨｏｒは、以下の条件（斜体で強調）に起因して偽であるように設定される。

６．４．２　許可される２分割プロセス

．．．

－　それ以外の場合に、以下の条件の全てが真である場合、ａｌｌｏｗＢｔＳｐｌｉｔは、ＦＡＬＳＥに等しく設定される。

－　ｂｔＳｐｌｉｔが、ＳＰＬＩＴ＿ＢＴ＿ＨＯＲに等しい。

－　ｃｂＷｉｄｔｈが、６４より大きい。

－　ｙ０＋ｃｂＨｅｉｇｈｔが、ｐｉｃ＿ｈｅｉｇｈｔ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きい。

－　それ以外の場合に、以下の条件の全てが真である場合、ａｌｌｏｗＢｔＳｐｌｉｔは、ＦＡＬＳＥに等しく設定される。

－　ｘ０＋ｃｂＷｉｄｔｈが、ｐｉｃ＿ｗｉｄｔｈ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きい。

－　ｙ０＋ｃｂＨｅｉｇｈｔが、ｐｉｃ＿ｈｅｉｇｈｔ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐ

ｌｅｓより大きい。
　－ ｃｂＷｉｄｔｈが、ｍｉｎＱｔＳｉｚｅより大きい。
－ それ以外の場合に、以下の条件の全てが真である場合、ａｌｌｏｗＢｔＳｐｌｉｔは、ＦＡＬＳＥに等しく設定される。
　－ ｂｔＳｐｌｉｔが、ＳＰＬＩＴ＿ＢＴ＿ＨＯＲに等しい。
　－ ｘ０＋ｃｂＷｉｄｔｈが、ｐｉｃ＿ｗｉｄｔｈ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きい。
　－ ｙ０＋ｃｂＨｅｉｇｈｔが、ｐｉｃ＿ｈｅｉｇｈｔ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓ以下である。
．．．
【００９９】
［００１２０］ 変数ａｌｌｏｗＳｐｌｉｔＢｔＶｅｒは、以下の条件（斜体で強調）に起因して偽であるように設定される。
６．４．２ 許可される２分割プロセス
．．．
－ それ以外の場合に、以下の条件の全てが真である場合、ａｌｌｏｗＢｔＳｐｌｉｔは、ＦＡＬＳＥに等しく設定される。
　－ ｂｔＳｐｌｉｔが、ＳＰＬＩＴ＿ＢＴ＿ＶＥＲに等しい。
　－ ｙ０＋ｃｂＨｅｉｇｈｔが、ｐｉｃ＿ｈｅｉｇｈｔ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きい。
－ それ以外の場合に、以下の条件の全てが真である場合、ａｌｌｏｗＢｔＳｐｌｉｔは、ＦＡＬＳＥに等しく設定される。
　－ ｂｔＳｐｌｉｔが、ＳＰＬＩＴ＿ＢＴ＿ＶＥＲに等しい。
　－ ｃｂＨｅｉｇｈｔが、６４より大きい。
　－ ｘ０＋ｃｂＷｉｄｔｈが、ｐｉｃ＿ｗｉｄｔｈ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きい。
．．．
【０１００】
［００１２１］ 変数ａｌｌｏｗＳｐｌｉｔＴｔＨｏｒ及びａｌｌｏｗＳｐｌｉｔＴｔＶｅｒは、以下の条件（斜体で強調）に起因して、偽であるように設定される。
６．４．３ 許可される３分割プロセス
．．．
－ 以下の条件のうちの１つ以上が真である場合、ａｌｌｏｗＴｔＳｐｌｉｔは、ＦＡＬＳＥに等しく設定される。
　－ ｃｂＳｉｚｅが、２×ＭｉｎＴｔＳｉｚｅＹ以下である。
　－ ｃｂＷｉｄｔｈが、Ｍｉｎ（６４，ｍａｘＴｔＳｉｚｅ）より大きい。
　－ ｃｂＨｅｉｇｈｔが、Ｍｉｎ（６４，ｍａｘＴｔＳｉｚｅ）より大きい。
　－ ｍｔｔＤｅｐｔｈが、ｍａｘＭｔｔＤｅｐｔｈ以上である。
　－ ｘ０＋ｃｂＷｉｄｔｈが、ｐｉｃ＿ｗｉｄｔｈ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きい。
　－ ｙ０＋ｃｂＨｅｉｇｈｔが、ｐｉｃ＿ｈｅｉｇｈｔ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きい。
．．．
【０１０１】
［００１２２］ これらの変数全てが偽に設定されるとき、ＣＵレベルにおいて分割フラグがシグナリングされないことがある。構文要素ｓｐｌｉｔ＿ｃｕ＿ｆｌａｇが、１であると推定され、構文要素ｓｐｌｉｔ＿ｑｔ＿ｆｌａｇが０であると推定され、構文要素ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｖｅｒｔｉｃａｌ＿ｆｌａｇが１であると推定され、構文要素ｍｔｔ＿ｓｐｌｉｔ＿ｃｕ＿ｂｉｎａｒｙ＿ｆｌａｇが０であると推定される。この場合、ブロックは、ＳＰＬＩＴ＿ＴＴ＿ＶＥＲを用いて分割されてもよく、それは、ＶＰＤＵの制

約に違反し得る。
【０１０２】
［00123］全てのツリー分割モードが許可されない第２の例示的な場合として、ピクチャ境界に位置し、且つピクチャ境界を超えるサンプルを含むブロックに対して全てのツリー分割が許可されない。最小ＱＴサイズが最小ＣＵサイズ（前の表の構文要素ｌｏｇ２＿ｍｉｎ＿ｌｕｍａ＿ｃｏｄｉｎｇ＿ｂｌｏｃｋ＿ｓｉｚｅ＿ｍｉｎｕｓ２）より大きく、最大ＢＴ／ＴＴ深さ（前の表の構文要素ｓｐｓ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｅｒ＿ｓｌｉｃｅ、ｓｐｓ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａ、ｓｐｓ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａ、ｐｉｃ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｅｒ＿ｓｌｉｃｅ、ｐｉｃ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａ、及びｐｉｃ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａ）が０に等しいとき、変数ａｌｌｏｗＳｐｌｉｔＱｔ、ａｌｌｏｗＳｐｌｉｔＢｔＨｏｒ、ａｌｌｏｗＳｐｌｉｔＢｔＶｅｒ、ａｌｌｏｗＳｐｌｉｔＴｔＨｏｒ、及びａｌｌｏｗＳｐｌｉｔＴｔＶｅｒの全てが偽に設定される。
【０１０３】
［00124］図１８は、本開示のいくつかの実施形態に係る、ＱＴ、ＢＴ、又はＢＴ分割のいずれもピクチャ境界で許可されない例示的ブロックを示す。ＣＴＵ（例えば、ＣＴＵ１８０１、ＣＴＵ１８０３、又はＣＴＵ１８０５）が、４つの６４×６４ブロックに分割する４分木を用いた第１の分割である。次いで、６４×６４ブロックのそれぞれは、さらに分割することができない。しかしながら、灰色で示されたブロックの部分は、右及び／又は下のピクチャ境界を超え、それはＶＶＣ設計において許可されない。灰色で示されたブロックについて、変数ａｌｌｏｗＳｐｌｉｔＱｔは、以下の条件（斜体で強調）に起因して偽に設定される。
６．４．１　許可される４分割プロセス
．．．
－　以下の条件のうちの１つ以上が真である場合、ａｌｌｏｗＳｐｌｉｔＱｔは、ＦＡＬＳＥに等しく設定される。
　－　ｔｒｅｅＴｙｐｅが、ＳＩＮＧＬＥ＿ＴＲＥＥ又はＤＵＡＬ＿ＴＲＥＥ＿ＬＵＭＡに等しく、且つｃｂＳｉｚｅが、ＭｉｎＱｔＳｉｚｅＹ以下である。
　－　ｔｒｅｅＴｙｐｅが、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しく、且つｃｂＳｉｚｅ／ＳｕｂＷｉｄｔｈＣが、ＭｉｎＱｔＳｉｚｅＣ以下である。
．．．
【０１０４】
［00125］灰色で示されたブロックについて、変数ａｌｌｏｗＳｐｌｉｔＢｔＨｏｒ及びａｌｌｏｗＳｐｌｉｔＢｔＶｅｒは、以下の条件（斜体で強調）に起因して偽に設定される。
６．４．２　許可される２分割プロセス
．．．
【０１０５】
［00126］変数ａｌｌｏｗＢｔＳｐｌｉｔは、以下のように導出される。
－　以下の条件の１つ以上が真である場合、ａｌｌｏｗＢｔＳｐｌｉｔは、ＦＡＬＳＥに等しく設定される。
　－　ｃｂＳｉｚｅが、ＭｉｎＢｔＳｉｚｅＹ以下である。
　－　ｃｂＷｉｄｔｈが、ｍａｘＢｔＳｉｚｅより大きい。
　－　ｃｂＨｅｉｇｈｔが、ｍａｘＢｔＳｉｚｅより大きい。
　－　ｍｔｔＤｅｐｔｈが、ｍａｘＭｔｔＤｅｐｔｈ以上である。
．．．
【０１０６】

[00127] 灰色で示されたブロックについて、変数ａｌｌｏｗＳｐｌｉｔＴｔＨｏｒ及びａｌｌｏｗＳｐｌｉｔＴｔＶｅｒは、以下の条件（斜体で強調）に起因して偽に設定される。

６．４．３　許可される３分割プロセス

．．．

【０１０７】

[00128] 変数ａｌｌｏｗＴｔＳｐｌｉｔは、以下のように導出される。

－　以下の条件のうちの１つ以上が真である場合、ａｌｌｏｗＴｔＳｐｌｉｔは、ＦＡＬＳＥに等しく設定される。

－　ｃｂＳｉｚｅが、２×ＭｉｎＴｔＳｉｚｅＹ以下である。

－　ｃｂＷｉｄｔｈが、Ｍｉｎ（６４，ｍａｘＴｔＳｉｚｅ）より大きい。

－　ｃｂＨｅｉｇｈｔが、Ｍｉｎ（６４，ｍａｘＴｔＳｉｚｅ）より大きい。

－　ｍｔｔＤｅｐｔｈが、ｍａｘＭｔｔＤｅｐｔｈ以上である。

．．．

【０１０８】

[00129] 全てのツリー分割モードが許可されない第１の例示的な場合において説明されるように、現在のＶＶＣドラフト７において、ＣＵはピクチャ境界の外にサンプルを含み得るが、いくつかの条件下ではそれ以上ＣＵを分割できない。本開示の実施形態によっては、ＶＶＣにおけるＱＴ分割条件は変更され得る。ある態様では、ピクチャ境界の外にサンプルを含み、その幅又は高さがＮ（例えば、Ｎ＝１２８）に等しいブロックについて、最小ＱＴサイズがＮ（例えば１２８）より小さいときにＱＴ分割が用いられる。加えて、実施形態によっては、最小ＱＴサイズがＮ（例えば１２８）に等しいときにも、ＱＴ分割が用いられ得る。別の態様では、ＱＴ分割を用いることは、ＢＴ又はＴＴ分割を用いることよりも簡単であり得る。図１９は、本開示のいくつかの実施形態に係る、ＢＴ及びＴＴ分割を用いる例を示す概略図である。ブロックを分割するために複数のステップを必要とし得る。さらに、異なる位置に位置するブロックに対して分割が異なることがあり、それによって複雑になり得る。例えば、ブロック１９０３について、ＳＰＬＩＴ＿ＢＴ＿ＨＯＲ、ＳＰＬＩＴ＿ＢＴ＿ＨＯＲ、ＳＰＬＩＴ＿ＴＴ＿ＶＥＲ、ＳＰＬＩＴ＿ＴＴ＿ＶＥＲ、及びＳＰＬＩＴ＿ＴＴ＿ＶＥＲが、順次実行される。

【０１０９】

[00130] 実施形態によっては、ブロック区分のいずれも許可されないときに、４分木分割を用いることができ、構文要素ｓｐｌｉｔ＿ｑｔ＿ｆｌａｇは１であると推定され得る。構文要素ｓｐｌｉｔ＿ｑｔ＿ｆｌａｇは、符号化ユニットが半分の水平及び垂直サイズを有する符号化ユニットに分割されるかどうかを指定する。

【０１１０】

[00131] 構文要素ｓｐｌｉｔ＿ｑｔ＿ｆｌａｇが存在しないとき、以下（斜体で強調）を適用する。

－　以下の条件の全てが真である場合、ｓｐｌｉｔ＿ｑｔ＿ｆｌａｇは、１に等しいと推定される。

－　ｓｐｌｉｔ＿ｃｕ＿ｆｌａｇが、１に等しい。

－　ａｌｌｏｗＳｐｌｉｔＱｔ、ａｌｌｏｗＳｐｌｉｔＢｔＨｏｒ、ａｌｌｏｗＳｐｌｉｔＢｔＶｅｒ、ａｌｌｏｗＳｐｌｉｔＴｔＨｏｒ、及びａｌｌｏｗＳｐｌｉｔＴｔＶｅｒがＦＡＬＳＥに等しい。

－　それ以外の場合に、ａｌｌｏｗＳｐｌｉｔＱｔがＴＲＵＥに等しい場合、ｓｐｌｉｔ＿ｑｔ＿ｆｌａｇの値は、１に等しいと推定される。

－　それ以外の場合、ｓｐｌｉｔ＿ｑｔ＿ｆｌａｇの値は、０に等しいと推定される。

【０１１１】

[00132] 実施形態によっては、最小ＱＴサイズ制約は、ピクチャ境界に位置するブロックに適用することができない。ブロックの一部が下又は右ピクチャ境界を超えるとき、ブロックは、４分木を用いて分割され得る。許可される４分割プロセスが、以下のように

記載される。
６．４．１　許可される４分割プロセス
　本プロセスへの入力は、以下の通りである。
　－　ルマサンプル内の符号化ブロックサイズｃｂＳｉｚｅ
　－　マルチタイプツリー深さｍｔｔＤｅｐｔｈ
　－　符号化ツリーノードを区分するためにシングルツリー（ＳＩＮＧＬＥ＿ＴＲＥＥ）が用いられるか、又はデュアルツリーが用いられるか、及びデュアルツリーが用いられるときに、ルマ成分（ＤＵＡＬ＿ＴＲＥＥ＿ＬＵＭＡ）が現在処理されるか又はクロマ成分（ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡ）が現在処理されるかを指定する、変数ｔｒｅｅＴｙｐｅ
　－　符号化ツリーノードの内側の符号化ユニットに対して、イントラ符号化モード（ＭＯＤＥ＿ＩＮＴＲＡ）、ＩＢＣ符号化モード（ＭＯＤＥ＿ＩＢＣ）、及びインター符号化モードが用いられ得るか（ＭＯＤＥ＿ＴＹＰＥ＿ＡＬＬ）、又はイントラ符号化モード及びＩＢＣ符号化モードのみが用いられ得るか（ＭＯＤＥ＿ＴＹＰＥ＿ＩＮＴＲＡ）、又はインター符号化モードのみが用いられ得るか（ＭＯＤＥ＿ＴＹＰＥ＿ＩＮＴＥＲ）を指定する、変数ｍｏｄｅＴｙｐｅ
　本プロセスの出力は、変数ａｌｌｏｗＳｐｌｉｔＱｔである。
【０１１２】
［00133］変数ａｌｌｏｗＳｐｌｉｔＱｔは、以下（斜体で強調）のように導出される。
　－　以下の条件の全てが真である場合、ａｌｌｏｗＳｐｌｉｔＱｔは、ＴＲＵＥに等しく設定される。
　　－　ｔｒｅｅＴｙｐｅが、ＳＩＮＧＬＥ＿ＴＲＥＥ又はＤＵＡＬ＿ＴＲＥＥ＿ＬＵＭＡに等しい。
　　－　ｃｂＳｉｚｅが、１２８に等しい。
　　－　ＭｉｎＱｔＳｉｚｅＹが、１２８に等しい。
　　－　ｘ０＋ｃｂＷｉｄｔｈが、ｐｉｃ＿ｗｉｄｔｈ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きいか、又はｙ０＋ｃｂＨｅｉｇｈｔが、ｐｉｃ＿ｈｅｉｇｈｔ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きい。
　－　それ以外の場合に、以下の条件の全てが真である場合、ａｌｌｏｗＳｐｌｉｔＱｔは、ＴＲＵＥに等しく設定される。
　　－　ｔｒｅｅＴｙｐｅが、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しい。
　　－　ＣｂＳｉｚｅ／ＳｕｂＷｉｄｔｈＣが、１２８に等しい。
　　－　ＭｉｎＱｔＳｉｚｅＣが、１２８に等しい。
　　－　ｘ０＋ｃｂＷｉｄｔｈが、ｐｉｃ＿ｗｉｄｔｈ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きいか、又はｙ０＋ｃｂＨｅｉｇｈｔが、ｐｉｃ＿ｈｅｉｇｈｔ＿ｉｎ＿ｌｕｍａ＿ｓａｍｐｌｅｓより大きい。
　－　それ以外の場合に、以下の条件のうちの１つ以上が真である場合、ａｌｌｏｗＳｐｌｉｔＱｔは、ＦＡＬＳＥに等しく設定される。
　　－　ｔｒｅｅＴｙｐｅが、ＳＩＮＧＬＥ＿ＴＲＥＥ又はＤＵＡＬ＿ＴＲＥＥ＿ＬＵＭＡに等しく、且つｃｂＳｉｚｅが、ＭｉｎＱｔＳｉｚｅＹ以下である。
　　－　ｔｒｅｅＴｙｐｅが、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しく、且つｃｂＳｉｚｅ／ＳｕｂＷｉｄｔｈＣが、ＭｉｎＱｔＳｉｚｅＣ以下である。
　　－　ｍｔｔＤｅｐｔｈが、０に等しくない。
　　－　ｔｒｅｅＴｙｐｅが、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しく、且つ（ｃｂＳｉｚｅ／ＳｕｂＷｉｄｔｈＣ）が、４以下である。
　　－　ｔｒｅｅＴｙｐｅが、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しく、且つｍｏｄｅＴｙｐｅが、ＭＯＤＥ＿ＴＹＰＥ＿ＩＮＴＲＡに等しい。
　－　それ以外の場合、ａｌｌｏｗＳｐｌｉｔＱｔは、ＴＲＵＥに等しく設定される。
【０１１３】
［00134］実施形態によっては、ビットストリーム適合性が、最小ＱＴサイズの構文に

追加され得る。最小ＱＴサイズが６４以下である必要があることを要し得る。
【０１１４】
[00135] 構文要素ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａは、ＣＴＵ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における最小サイズの底を２とする対数と、ＳＰＳを参照する２（Ｉ）に等しいｓｌｉｃｅ＿ｔｙｐｅを有するスライスのルマＣＵについてのルマサンプルにおける最小符号化ブロックサイズの底を２とする対数との間の、デフォルト差を指定する。構文要素ｐａｒｔｉｔｉｏｎ＿ｃｏｎｓｔｒａｉｎｔｓ＿ｏｖｅｒｒｉｄｅ＿ｅｎａｂｌｅｄ＿ｆｌａｇが１に等しいとき、デフォルト差は、ＳＰＳを参照するＰＨに存在する構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｌｕｍａによってオーバーライドされ得る。構文要素ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａの値は、０からＣｔｂＬｏｇ２ＳｉｚｅＹ－ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹの範囲（両端を含む）にある。ＣＴＵ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における最小サイズの底を２とする対数は、以下のように（斜体で強調）導出される。

$$MinQtLog2SizeIntraY = sps_log2_diff_min_qt_min_cb_intra_slice_luma + MinCbLog2SizeY$$

$$VSize = Min(64, CtbSizeY)$$

【０１１５】
[00136] 実施形態によっては、（１＜＜ＭｉｎＱｔＬｏｇ２ＳｉｚｅＩｎｔｒａＹ）の値がＶＳｉｚｅ以下であることが、ビットストリーム適合性の要件であり得る。
【０１１６】
[00137] 構文要素ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｅｒ＿ｓｌｉｃｅは、ＣＴＵ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における最小サイズの底を２とする対数と、ＳＰＳを参照する０（Ｂ）又は１（Ｐ）に等しいｓｌｉｃｅ＿ｔｙｐｅを有するスライスのルマＣＵについてのルマサンプルにおける最小ルマ符号化ブロックサイズの底を２とする対数との間のデフォルト差を指定する。構文要素ｐａｒｔｉｔｉｏｎ＿ｃｏｎｓｔｒａｉｎｔｓ＿ｏｖｅｒｒｉｄｅ＿ｅｎａｂｌｅｄ＿ｆｌａｇが１に等しいとき、デフォルト差は、ＳＰＳを参照するＰＨに存在する構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｌｕｍａによってオーバーライドされ得る。構文要素ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｅｒ＿ｓｌｉｃｅの値は、０からＣｔｂＬｏｇ２ＳｉｚｅＹ－ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹの範囲（両端を含む）にある。ＣＴＵ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における最小サイズの底を２とする対数は、以下のように（斜体で強調）導出される。

$$MinQtLog2SizeInterY = sps_log2_diff_min_qt_min_cb_inter_slice + MinCbLog2SizeY$$

$$VSize = Min(64, CtbSizeY)$$

【０１１７】
[00138] 実施形態によっては、（１＜＜ＭｉｎＱｔＬｏｇ２ＳｉｚｅＩｎｔｅｒＹ）の値がＶＳｉｚｅ以下であることが、ビットストリーム適合性の要件であり得る。
【０１１８】
[00139] 構文要素ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａは、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しいｔｒｅｅＴｙｐｅを有するクロマＣＴＵの４分木分割から生じるクロマリーフブロックのルマサンプルにおける最小サイズの底を２とする対数と、ＳＰＳを参照する２（Ｉ）に等しいｓｌｉｃｅ＿ｔｙｐｅを有するスライスにおけるＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しいｔｒｅｅＴｙｐｅを有するクロマＣＵについてのルマサンプルにおける最小符号化ブロックサイズの底を２とする対数との間のデフォルト差を指定する。構文要素ｐａｒｔｉｔｉｏｎ＿ｃｏｎｓｔｒａｉｎｔｓ＿ｏｖｅｒｒｉｄｅ＿ｅｎａｂｌｅｄ＿ｆｌａｇが１

に等しいとき、デフォルト差は、ＳＰＳを参照するＰＨに存在する構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｃｈｒｏｍａによってオーバーライドされ得る。構文要素ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａの値は、０からＣｔｂＬｏｇ２ＳｉｚｅＹ－ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹの範囲（両端を含む）にある。存在しない場合、構文要素ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａの値は、０に等しいと推定される。ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しいｔｒｅｅＴｙｐｅを有するＣＴＵ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における最小サイズの底を２とする対数は、以下（斜体で強調）のように導出される。

ＭｉｎＱｔＬｏｇ２ＳｉｚｅＩｎｔｒａＣ＝ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａ＋ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ

ＶＳｉｚｅ＝Ｍｉｎ（６４，ＣｔｂＳｉｚｅＹ）

【０１１９】

[00140] 実施形態によっては、（１＜＜ＭｉｎＱｔＬｏｇ２ＳｉｚｅＩｎｔｒａＣ）の値がＶＳｉｚｅ以下であることが、ビットストリーム適合性の要件であり得る。

【０１２０】

[00141] 構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａは、ＣＴＵ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における最小サイズの底を２とする対数と、ＰＨに関連付けられた２（Ｉ）に等しいｓｌｉｃｅ＿ｔｙｐｅを有するスライスのルマＣＵについてのルマサンプルにおける最小符号化ブロックサイズの底を２とする対数との間の差を指定する。構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａの値は、０からＣｔｂＬｏｇ２ＳｉｚｅＹ－ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹの範囲（両端を含む）にある。存在しない場合、構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｌｕｍａの値は、構文要素ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａに等しいと推定される。実施形態によっては、（１＜＜（ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａ＋ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ））の値がＭｉｎ（６４，ＣｔｂＳｉｚｅＹ）以下であることが、ビットストリーム適合性の要件であり得る。

【０１２１】

[00142] 構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｅｒ＿ｓｌｉｃｅは、ＣＴＵ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における最小サイズの底を２とする対数と、ＰＨに関連付けられた０（Ｂ）又は１（Ｐ）に等しいｓｌｉｃｅ＿ｔｙｐｅを有するスライスのルマＣＵについてのルマサンプルにおける最小ルマ符号化ブロックサイズの底を２とする対数との間の差を指定する。構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｅｒ＿ｓｌｉｃｅの値は、０からＣｔｂＬｏｇ２ＳｉｚｅＹ－ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹの範囲（両端を含む）にある。存在しない場合、構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｌｕｍａの値は、構文要素ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｅｒ＿ｓｌｉｃｅに等しいと推定される。実施形態によっては、（１＜＜（ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｅｒ＿ｓｌｉｃｅ＋ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ））の値がＭｉｎ（６４，ＣｔｂＳｉｚｅＹ）以下であることが、ビットストリーム適合性の要件であり得る。

【０１２２】

[00143] 構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａは、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しいｔｒｅｅＴｙｐｅを有するクロマＣＴＵの４分木分割から生じるクロマリーフブロックのルマサンプルにおける最小サイズの底を２とする対数と、ＰＨに関連付けられた２（Ｉ）に等しいｓｌｉｃｅ＿ｔｙｐｅを有するスライスにおけるＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡ

に等しいｔｒｅｅＴｙｐｅを有するクロマＣＵについてのルマサンプルにおける最小符号化ブロックサイズの底を２とする対数との間の差を指定する。構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａの値は、０からＣｔｂＬｏｇ２ＳｉｚｅＹ－ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹの範囲（両端を含む）にある。存在しない場合、構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａの値は、構文要素ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａに等しいと推定される。実施形態によっては、（１＜＜（ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａ＋ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ））の値がＭｉｎ（６４，ＣｔｂＳｉｚｅＹ）以下であることが、ビットストリーム適合性の要件であり得る。

【０１２３】

[00144] 実施形態によっては、全てのツリー分割モードが許可されない上記第２の例示的な場合において、ブロック区分のいずれも許可されないとき、４分木分割が用いられてもよく、構文要素ｓｐｌｉｔ＿ｑｔ＿ｆｌａｇが１であると推定され得る。

【０１２４】

[00145] 構文要素ｓｐｌｉｔ＿ｑｔ＿ｆｌａｇは、符号化ユニットが、半分の水平及び垂直サイズを有する符号化ユニットに分割されるかどうかを指定する。構文要素ｓｐｌｉｔ＿ｑｔ＿ｆｌａｇが存在しないとき、以下（斜体で強調）を適用する。

－ 以下の条件の全てが真である場合、ｓｐｌｉｔ＿ｑｔ＿ｆｌａｇは、１に等しいと推定される。

－ ｓｐｌｉｔ＿ｃｕ＿ｆｌａｇが、１に等しい。

－ ａｌｌｏｗＳｐｌｉｔＱｔ、ａｌｌｏｗＳｐｌｉｔＢｔＨｏｒ、ａｌｌｏｗＳｐｌｉｔＢｔＶｅｒ、ａｌｌｏｗＳｐｌｉｔＴｔＨｏｒ、及びａｌｌｏｗＳｐｌｉｔＴｔＶｅｒがＦＡＬＳＥに等しい。

－ それ以外の場合に、ａｌｌｏｗＳｐｌｉｔＱｔがＴＲＵＥに等しい場合、ｓｐｌｉｔ＿ｑｔ＿ｆｌａｇの値は、１に等しいと推定される。

－ それ以外の場合、ｓｐｌｉｔ＿ｑｔ＿ｆｌａｇの値は、０に等しいと推定される。

【０１２５】

[00146] 実施形態によっては、全てのツリー分割モードが許可されない、上記第２の例示的な場合において、ビットストリーム適合性が、最小ＱＴサイズ及び最大ＢＴ／ＴＴ深さの構文に追加され得る。

【０１２６】

[00147] 構文要素ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａは、ＣＴＵ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における最小サイズの底を２とする対数と、ＳＰＳを参照する２（Ｉ）に等しいｓｌｉｃｅ＿ｔｙｐｅを有するスライスのルマＣＵについてのルマサンプルにおける最小符号化ブロックサイズの底を２とする対数との間のデフォルト差を指定する。構文要素ｐａｒｔｉｔｉｏｎ＿ｃｏｎｓｔｒａｉｎｔｓ＿ｏｖｅｒｒｉｄｅ＿ｅｎａｂｌｅｄ＿ｆｌａｇが１に等しいとき、デフォルト差は、ＳＰＳを参照するＰＨに存在する構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｌｕｍａによってオーバーライドされ得る。構文要素ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａの値は、０からＣｔｂＬｏｇ２ＳｉｚｅＹ－ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹの範囲（両端を含む）にある。ＣＴＵ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における最小サイズの底を２とする対数は、以下のように（斜体で強調）導出される。

ＭｉｎＱｔＬｏｇ２ＳｉｚｅＩｎｔｒａＹ＝ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａ＋ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ

【０１２７】

[00148] 構文要素ｓｐｓ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎ

ｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａは、ＳＰＳを参照する２（Ｉ）に等しいｓｌｉｃｅ＿ｔｙｐｅを有するスライスにおける４分木リーフのマルチタイプツリー分割から生じる符号化ユニットについてのデフォルト最大階層深さを指定する。構文要素ｐａｒｔｉｔｉｏｎ＿ｃｏｎｓｔｒａｉｎｔｓ＿ｏｖｅｒｒｉｄｅ＿ｅｎａｂｌｅｄ＿ｆｌａｇが１に等しいとき、デフォルト最大階層深さは、ＳＰＳを参照するＰＨに存在する構文要素ｐｉｃ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａによってオーバーライドされ得る。構文要素ｓｐｓ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａの値は、０から２×（ＣｔｂＬｏｇ２ＳｉｚｅＹ－ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ）の範囲（両端を含む）にある。実施形態によっては、（ＭｉｎＱｔＬｏｇ２ＳｉｚｅＩｎｔｒａＹ－ｓｐｓ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａ／２）の値がＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ以下であることが、ビットストリーム適合性の要件であり得る。

【０１２８】

[00149] 構文要素ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｅｒ＿ｓｌｉｃｅは、ＣＴＵ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における最小サイズの底を２とする対数と、ＳＰＳを参照する０（Ｂ）又は１（Ｐ）に等しいｓｌｉｃｅ＿ｔｙｐｅを有するスライスのルマＣＵについてのルマサンプルにおける最小ルマ符号化ブロックサイズの底を２とする対数との間のデフォルト差を指定する。構文要素ｐａｒｔｉｔｉｏｎ＿ｃｏｎｓｔｒａｉｎｔｓ＿ｏｖｅｒｒｉｄｅ＿ｅｎａｂｌｅｄ＿ｆｌａｇが１に等しいとき、デフォルト差は、ＳＰＳを参照するＰＨに存在する構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｌｕｍａによってオーバーライドされ得る。構文要素ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｅｒ＿ｓｌｉｃｅの値は、０からＣｔｂＬｏｇ２ＳｉｚｅＹ－ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹの範囲（両端を含む）にある。ＣＴＵ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における最小サイズの底を２とする対数は、以下のように導出される。

ＭｉｎＱｔＬｏｇ２ＳｉｚｅＩｎｔｅｒＹ＝ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｅｒ＿ｓｌｉｃｅ＋ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ

【０１２９】

[00150] 構文要素ｓｐｓ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｅｒ＿ｓｌｉｃｅは、ＳＰＳを参照する０（Ｂ）又は１（Ｐ）に等しいｓｌｉｃｅ＿ｔｙｐｅを有するスライスにおける４分木リーフのマルチタイプツリー分割から生じる符号化ユニットについてのデフォルト最大階層深さを指定する。構文要素ｐａｒｔｉｔｉｏｎ＿ｃｏｎｓｔｒａｉｎｔｓ＿ｏｖｅｒｒｉｄｅ＿ｅｎａｂｌｅｄ＿ｆｌａｇが１に等しいとき、デフォルト最大階層深さは、ＳＰＳを参照するＰＨに存在する構文要素ｐｉｃ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｅｒ＿ｓｌｉｃｅによってオーバーライドされ得る。構文要素ｓｐｓ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｅｒ＿ｓｌｉｃｅの値は、０から２×（ＣｔｂＬｏｇ２ＳｉｚｅＹ－ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ）の範囲（両端を含む）にある。実施形態によっては、（ＭｉｎＱｔＬｏｇ２ＳｉｚｅＩｎｔｅｒＹ－ｓｐｓ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｅｒ＿ｓｌｉｃｅ／２）の値がＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ以下であることが、ビットストリーム適合性の要件であり得る。

【０１３０】

[00151] 構文要素ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａは、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しいｔｒｅｅＴｙｐｅを有するクロマＣＴＵの４分木分割から生じるクロマリーフブロックのルマサンプルにおける最小サイズの底を２とする対数と、ＳＰＳを参照する２（Ｉ）に等しいｓｌｉｃｅ＿ｔｙｐｅを有するスライスのＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しいｔｒｅｅＴｙｐｅを有するクロマＣＵについてのルマサンプルにおける最小符号化ブロックサイズの底を２とする対数との間のデフォルト差を指定する。構文要素ｐａｒｔｉｔｉｏ

ｎ＿ｃｏｎｓｔｒａｉｎｔｓ＿ｏｖｅｒｒｉｄｅ＿ｅｎａｂｌｅｄ＿ｆｌａｇが１に等しいとき、デフォルト差は、ＳＰＳを参照するＰＨに存在する構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｃｈｒｏｍａによってオーバーライドされ得る。構文要素ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａの値は、０からＣｔｂＬｏｇ２ＳｉｚｅＹ－ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹの範囲（両端を含む）にある。存在しない場合、構文要素ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａの値は、０に等しいと推定される。ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しいｔｒｅｅＴｙｐｅを有するＣＴＵ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における最小サイズの底を２とする対数は、以下のように導出される。

ＭｉｎＱｔＬｏｇ２ＳｉｚｅＩｎｔｒａＣ＝ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａ＋ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ

【０１３１】

[00152] 構文要素ｓｐｓ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａは、ＳＰＳを参照する２（Ｉ）に等しいｓｌｉｃｅ＿ｔｙｐｅを有するスライスにおける、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しいｔｒｅｅＴｙｐｅを有するクロマ４分木リーフのマルチタイプツリー分割から生じるクロマ符号化ユニットについてのデフォルト最大階層深さを指定する。構文要素ｐａｒｔｉｔｉｏｎ＿ｃｏｎｓｔｒａｉｎｔｓ＿ｏｖｅｒｒｉｄｅ＿ｅｎａｂｌｅｄ＿ｆｌａｇが１に等しいとき、デフォルト最大階層深さは、ＳＰＳを参照するＰＨに存在する構文要素ｐｉｃ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｃｈｒｏｍａによってオーバーライドされ得る。構文要素ｓｐｓ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａの値は、０から２×（ＣｔｂＬｏｇ２ＳｉｚｅＹ－ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ）の範囲（両端を含む）にある。存在しない場合、構文要素ｓｐｓ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａの値は、０に等しいと推定される。実施形態によっては、（ＭｉｎＱｔＬｏｇ２ＳｉｚｅＩｎｔｒａＣ－ｓｐｓ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａ／２）の値がＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ以下であることが、ビットストリーム適合性の要件であり得る。

【０１３２】

[00153] 構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａは、ＣＴＵ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における最小サイズの底を２とする対数と、ＰＨに関連付けられた２（Ｉ）に等しいｓｌｉｃｅ＿ｔｙｐｅを有するスライスにおけるルマＣＵについてのルマサンプルにおける最小符号化ブロックサイズの底を２とする対数との間の差を指定する。構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａの値は、０からＣｔｂＬｏｇ２ＳｉｚｅＹ－ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹの範囲（両端を含む）にある。存在しない場合、構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｌｕｍａの値は、構文要素ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａに等しいと推定される。

【０１３３】

[00154] 構文要素ｐｉｃ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａは、ＰＨに関連付けられた２（Ｉ）に等しいｓｌｉｃｅ＿ｔｙｐｅを有するスライスにおける４分木リーフのマルチタイプツリー分割から生じる符号化ユニットについての最大階層深さを指定する。構文要素ｐｉｃ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａの値は、０から２×（ＣｔｂＬｏｇ２ＳｉｚｅＹ－ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ）の範囲（両端を含む）にある。存在しない場合、構文要素ｐｉｃ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａの値は、構文要素ｓｐｓ＿ｍａｘ＿ｍｔｔ＿ｈ

ｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａに等しいと推定される。実施形態によっては、（ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａ＋ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ－ｐｉｃ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｌｕｍａ／２）の値がＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ以下であることが、ビットストリーム適合性の要件であり得る。

【０１３４】

［００１５５］構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｅｒ＿ｓｌｉｃｅは、ＣＴＵ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における最小サイズの底を２とする対数と、ＰＨに関連付けられた０（Ｂ）又は１（Ｐ）に等しいｓｌｉｃｅ＿ｔｙｐｅを有するスライスのルマＣＵについてのルマサンプルにおける最小ルマ符号化ブロックサイズの底を２とする対数との間の差を指定する。構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｅｒ＿ｓｌｉｃｅの値は、０からＣｔｂＬｏｇ２ＳｉｚｅＹ－ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹの範囲（両端を含む）にある。存在しない場合、構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｌｕｍａの値は、構文要素ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｅｒ＿ｓｌｉｃｅに等しいと推定される。

【０１３５】

［００１５６］構文要素ｐｉｃ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｅｒ＿ｓｌｉｃｅは、ＰＨに関連付けられた０（Ｂ）又は１（Ｐ）に等しいｓｌｉｃｅ＿ｔｙｐｅを有するスライスにおける４分木リーフのマルチタイプツリー分割から生じる符号化ユニットについての最大階層深さを指定する。構文要素ｐｉｃ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｅｒ＿ｓｌｉｃｅの値は、０から２×（ＣｔｂＬｏｇ２ＳｉｚｅＹ－ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ）の範囲（両端を含む）にある。存在しない場合、構文要素ｐｉｃ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｅｒ＿ｓｌｉｃｅの値は、構文要素ｓｐｓ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｅｒ＿ｓｌｉｃｅに等しいと推定される。実施形態によっては、（ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｅｒ＿ｓｌｉｃｅ＋ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ－ｐｉｃ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｅｒ＿ｓｌｉｃｅ／２）の値がＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ以下であることが、ビットストリーム適合性の要件であり得る。

【０１３６】

［００１５７］構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａは、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しいｔｒｅｅＴｙｐｅを有するクロマＣＴＵの４分木分割から生じるクロマリーフブロックのルマサンプルにおける最小サイズの底を２とする対数と、ＰＨに関連付けられた２（Ｉ）に等しいｓｌｉｃｅ＿ｔｙｐｅを有するスライスにおけるＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しいｔｒｅｅＴｙｐｅを有するクロマＣＵについてのルマサンプルにおける最小符号化ブロックサイズの底を２とする対数との間の差を指定する。構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａの値は、０からＣｔｂＬｏｇ２ＳｉｚｅＹ－ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹの範囲（両端を含む）にある。存在しない場合、構文要素ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａの値は、構文要素ｓｐｓ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａに等しいと推定される。

【０１３７】

［００１５８］構文要素ｐｉｃ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａは、ＰＨに関連付けられた２（Ｉ）に等しいｓｌｉｃｅ＿ｔｙｐｅを有するスライスにおける、ＤＵＡＬ＿ＴＲＥＥ＿ＣＨＲＯＭＡに等しいｔｒｅｅＴｙｐｅを有するクロマ４分木リーフのマルチタイプツリー分割から生じるクロマ

符号化ユニットについての最大階層深さを指定する。構文要素ｐｉｃ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａの値は、０から２×（ＣｔｂＬｏｇ２ＳｉｚｅＹ－ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ）の範囲（両端を含む）にある。存在しない場合、構文要素ｐｉｃ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａの値は、構文要素ｓｐｓ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａに等しいと推定される。実施形態によっては、（ｐｉｃ＿ｌｏｇ２＿ｄｉｆｆ＿ｍｉｎ＿ｑｔ＿ｍｉｎ＿ｃｂ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａ＋ＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ－ｐｉｃ＿ｍａｘ＿ｍｔｔ＿ｈｉｅｒａｒｃｈｙ＿ｄｅｐｔｈ＿ｉｎｔｒａ＿ｓｌｉｃｅ＿ｃｈｒｏｍａ／２）の値がＭｉｎＣｂＬｏｇ２ＳｉｚｅＹ以下であることが、ビットストリーム適合性の要件であり得る。

【０１３８】

［00159］図２０は、本開示のいくつかの実施形態に係る、例示的映像処理方法２０００のフローチャートを示す。方法２０００は、符号器（例えば、図２Ａのプロセス２００Ａ又は図２Ｂのプロセス２００Ｂによる）、復号器（例えば、図３Ａのプロセス３００Ａ又は図３Ｂのプロセス３００Ｂによる）によって実行されてもよく、装置（例えば、図４の装置４００）の１つ以上のソフトウェア又はハードウェアコンポーネントによって実行されてもよい。例えば、プロセッサ（例えば、図４のプロセッサ４０２）は、方法２０００を実行し得る。実施形態によっては、方法２０００は、コンピュータ（例えば、図４の装置４００）によって実行される、プログラムコードなどのコンピュータ実行可能命令を含むコンピュータ可読媒体において具現化される、コンピュータプログラム製品によって実施され得る。

【０１３９】

［00160］ステップ２００１において、符号化ブロックがピクチャ境界の外側にサンプルを含むかどうかについての決定が行われ得る。実施形態によっては、ピクチャ境界が、下ピクチャ境界又は右ピクチャ境界であってもよい。ピクチャ境界の外側にサンプルを含む符号化ブロックについての例として、図１７を参照すると、符号化ブロック１７０１は、ピクチャ１７００の右ピクチャ境界を超え、符号化ブロック１７０５は、ピクチャ１７００の下ピクチャ境界を超え、符号化ブロック１７０３は、ピクチャ１７００の下及び右ピクチャ境界の両方を超える。

【０１４０】

［00161］ステップ２００３において、符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されることに応じて、符号化ブロックは、ＱＴモードを用いて分割され得る。実施形態によっては、符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されることに応じて、方法２０００は、符号化ブロックを分割するためにＢＴモード及びＴＴモードが用いられることを許可されないと決定し得る。例えば、変数ａｌｌｏｗＳｐｌｉｔＢｔＨｏｒ、ａｌｌｏｗＳｐｌｉｔＢｔＶｅｒ、ａｌｌｏｗＳｐｌｉｔＴｔＨｏｒ、及びａｌｌｏｗＳｐｌｉｔＴｔＶｅｒは、ＦＡＬＳＥに等しいと決定され得る。

【０１４１】

［00162］実施形態によっては、符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されることに応じて、方法２０００は、符号化ブロックを含むビットストリームにＱＴフラグが存在するかどうかに関わらず、符号化ブロックがＱＴモードを用いて分割されるべきであると決定し得る。ＱＴフラグは、符号化ブロックがＱＴモードを用いて分割されるべきであるかどうかを示す。例えば、構文要素ｓｐｌｉｔ＿ｑｔ＿ｆｌａｇが存在しないとき、符号化ブロックがピクチャ境界の外側にサンプルを含み、且つ変数ａｌｌｏｗＳｐｌｉｔＱｔ、ａｌｌｏｗＳｐｌｉｔＢｔＨｏｒ、ａｌｌｏｗＳｐｌｉｔＢｔＶｅｒ、ａｌｌｏｗＳｐｌｉｔＴｔＨｏｒ、及びａｌｌｏｗＳｐｌｉｔＴｔＶｅｒがＦＡＬＳＥに等しいか、又はａｌｌｏｗＳｐｌｉｔＱｔがＴＲＵＥに等しい場合に、ｓｐｌｉｔ＿ｑｔ＿ｆｌａｇの値が１に等しいと推定され得る。

【０１４２】

[00163] 実施形態によっては、符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されることに応じて、方法２０００は、ＱＴモードが適用を許可される最小ブロックサイズに対するプリセット制約に関わらず、符号化ブロックを分割するためにＱＴモードが用いられることを許可されると決定し得る。例えば、最小ＱＴサイズ制約をピクチャ境界に位置する符号化ブロックに適用することはできない。

【０１４３】

[00164] 実施形態によっては、プリセット制約が、符号化ブロックに対するビットストリーム適合性を含み得る。ビットストリーム適合性は、ＱＴモードが適用を許可される最小ブロックサイズを設定し得る。例えば、最小ブロックサイズは、６４以下であるように設定され得る。ビットストリーム適合性は、最大ＢＴ深さ又は最大ＴＴ深さも設定し得る。

【０１４４】

[00165] 実施形態によっては、方法２０００は、符号化ブロックが分割されるべきであると決定することを含み得る。例えば、構文要素ｓｐｌｉｔ＿ｃｕ＿ｆｌａｇは、符号化ブロックが分割されるべきであるかどうかを示すために用いられ得る。構文要素ｓｐｌｉｔ＿ｃｕ＿ｆｌａｇが存在しないとき、それは、符号化ブロックが分割されるべきであることを示す１に等しいと推定され得る。

【０１４５】

[00166] 本開示の実施形態は、別の実施形態又はいくつかの他の実施形態と組み合わせ得ることを理解されたい。

【０１４６】

[00167] 実施形態は、以下の条項を用いてさらに記述することができる。

１．映像処理方法であって、

符号化ブロックがピクチャ境界の外側にサンプルを含むかどうかを決定することと、

符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されることに応じて、第１のパラメータの値に関わらず符号化ブロックの４分木分割を実行することであって、第１のパラメータが、符号化ブロックを分割するために４分木を用いることを許可されるかどうかを示すことと、

を含む、方法。

２．符号化ブロックの第１のフラグの値を決定することであって、第１のフラグが、符号化ブロックが複数のサブブロックに分割されるかどうかを示すことと、

符号化ブロックの第２のパラメータ、第３のパラメータ、第４のパラメータ、及び第５のパラメータの値を決定することであって、第２のパラメータ、第３のパラメータ、第４のパラメータ、及び第５のパラメータが、符号化ブロックを分割するために水平２分木、垂直２分木、水平３分木、及び垂直３分木を用いることを許可されるかどうかをそれぞれ示すことと、

をさらに含む、条項１に記載の方法。

３．第１のフラグの値が１に等しいこと、並びに第１のパラメータ、第２のパラメータ、第３のパラメータ、第４のパラメータ、及び第５のパラメータの値が０に等しいことに応じて、符号化ブロックの第２のフラグの値を１に設定することであって、第２のフラグが、符号化ブロックが４分木を用いて分割されるかどうかを示すことをさらに含む、条項２に記載の方法。

４．第１のパラメータの値が１に等しいことに応じて、符号化ブロックの第２のフラグの値を１に設定することであって、第２のフラグが、符号化ブロックが４分木を用いて分割されるかどうかを示すことをさらに含む、条項２に記載の方法。

５．符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されるとき、第１のフラグの値が１であるように設定することをさらに含む、条項２～４のいずれか一項に記載の方法。

６．映像処理装置であって、

命令を記憶するための少なくとも１つのメモリと、

少なくとも１つのプロセッサとを備え、少なくとも１つのプロセッサは、
符号化ブロックがピクチャ境界の外側にサンプルを含むかどうかを決定すること、及び
符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されることに応じて、第１のパラメータの値に関わらず符号化ブロックの４分木分割を実行することであって、第１のパラメータが、符号化ブロックを分割するために４分木を用いることを許可されるかどうかを示すことと、
を装置に実行させるために、命令を実行するように構成される、装置。
７．少なくとも１つのプロセッサが、
符号化ブロックの第１のフラグの値を決定することであって、第１のフラグが、符号化ブロックが複数のサブブロックに分割されるかどうかを示すことと、
符号化ブロックの第２のパラメータ、第３のパラメータ、第４のパラメータ、及び第５のパラメータの値を決定することであって、第２のパラメータ、第３のパラメータ、第４のパラメータ、及び第５のパラメータが、符号化ブロックを分割するために水平２分木、垂直２分木、水平３分木、及び垂直３分木を用いることを許可されるかどうかをそれぞれ示すことと、
を装置に実行させるために、命令を実行するように構成される、条項６に記載の装置。
８．少なくとも１つのプロセッサが、
第１のフラグの値が１に等しいこと、並びに第１のパラメータ、第２のパラメータ、第３のパラメータ、第４のパラメータ、及び第５のパラメータの値が０に等しいことに応じて、符号化ブロックの第２のフラグの値を１に設定することであって、第２のフラグが、符号化ブロックが４分木を用いて分割されるかどうかを示すこと
を装置に実行させるために、命令を実行するように構成される、条項７に記載の装置。
９．少なくとも１つのプロセッサが、
第１のパラメータの値が１に等しいことに応じて、符号化ブロックの第２のフラグの値を１に設定することであって、第２のフラグが、符号化ブロックが４分木を用いて分割されるかどうかを示すこと
を装置に実行させるために、命令を実行するように構成される、条項７に記載の装置。
１０．少なくとも１つのプロセッサが、
符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されるとき、第１のフラグの値が１であるように設定すること
を装置に実行させるために、命令を実行するように構成される、条項７～９のいずれか一項に記載の装置。
１１．命令のセット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、命令のセットが、
符号化ブロックがピクチャ境界の外側にサンプルを含むかどうかを決定することと、
符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されることに応じて、第１のパラメータの値に関わらず符号化ブロックの４分木分割を実行することであって、第１のパラメータが、符号化ブロックを分割するために４分木を用いることを許可されるかどうかを示すことと、
を映像処理装置に実行させるために、１つ以上の処理デバイスによって実行可能である、非一時的コンピュータ可読記憶媒体。
１２．命令のセットが、
符号化ブロックの第１のフラグの値を決定することであって、第１のフラグが、符号化ブロックが複数のサブブロックに分割されるかどうかを示す、決定することと、
符号化ブロックの第２のパラメータ、第３のパラメータ、第４のパラメータ、及び第５のパラメータの値を決定することであって、第２のパラメータ、第３のパラメータ、第４のパラメータ、及び第５のパラメータが、符号化ブロックを分割するために水平２分木、垂直２分木、水平３分木、及び垂直３分木を用いることを許可されるかどうかをそれぞれ示す、決定することと、
を映像処理装置に実行させるために１つ以上の処理デバイスによって実行可能である、

条項１１に記載の非一時的コンピュータ可読記憶媒体。
１３．命令のセットが、
第１のフラグの値が１に等しいこと、並びに第１のパラメータ、第２のパラメータ、第３のパラメータ、第４のパラメータ、及び第５のパラメータの値が０に等しいことに応じて、符号化ブロックの第２のフラグの値を１に設定することであって、第２のフラグが、符号化ブロックが４分木を用いて分割されるかどうかを示すこと
を映像処理装置に実行させるために、１つ以上の処理デバイスによって実行可能である、条項１２に記載の非一時的コンピュータ可読記憶媒体。
１４．命令のセットが、
第１のパラメータの値が１に等しいことに応じて、符号化ブロックの第２のフラグの値を１に設定することであって、第２のフラグが、符号化ブロックが４分木を用いて分割されるかどうかを示すこと
を映像処理装置に実行させるために、１つ以上の処理デバイスによって実行可能である、条項１２に記載の非一時的コンピュータ可読記憶媒体。
１５．符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されることに応じて、ＱＴモードを用いて符号化ブロックを分割することが、
符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されるとき、第１のフラグの値が１であるように設定することを含む、条項１２～１４のいずれか一項に記載の非一時的コンピュータ可読記憶媒体。
１６．映像処理方法であって、
符号化ブロックがピクチャ境界の外側にサンプルを含むかどうかを決定することと、
符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されることに応じて、４分木（ＱＴ）モードを用いて符号化ブロックを分割することと、を含む、方法。
１７．
符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されることに応じて、２分木（ＢＴ）モード及び３分木（ＴＴ）モードは、符号化ブロックを分割するために用いられることを許可されないと決定することをさらに含む、条項１６に記載の方法。
１８．符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されることに応じて、４分木（ＱＴ）モードを用いて符号化ブロックを分割することが、
ＱＴフラグが符号化ブロックを含むビットストリーム内に存在するかどうかに関わらず、ＱＴモードを用いて符号化ブロックを分割することであって、ＱＴフラグが、符号化ブロックがＱＴモードを用いて分割されるべきかどうかを示すことを含む、条項１６及び１７のいずれか一項に記載の方法。
１９．符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されることに応じて、４分木（ＱＴ）モードを用いて符号化ブロックを分割することが、
ＱＴモードの適用が許可される最小ブロックサイズに対するプリセット制約に関わらず、ＱＴモードを用いて符号化ブロックを分割することを含む、条項１６に記載の方法。
２０．プリセット制約が、符号化ブロックに関連付けられたビットストリーム適合性を含み、ビットストリーム適合性が、ＱＴモードの適用が許可される最小ブロックサイズを設定する、条項１９に記載の方法。
２１．最小ブロックサイズが、６４以下であるように設定される、条項２０に記載の方法。
２２．ビットストリーム適合性が、最大ＢＴ深さ又は最大ＴＴ深さを設定する、条項２０及び２１のいずれか一項に記載の方法。
２３．ピクチャ境界が、下ピクチャ境界又は右ピクチャ境界である、条項１６～２２のいずれか一項に記載の方法。
２４．映像処理装置であって、
命令を記憶するための少なくとも１つのメモリと、
少なくとも１つのプロセッサとを備え、少なくとも１つのプロセッサは、
符号化ブロックがピクチャ境界の外側にサンプルを含むかどうかを決定することと、

符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されることに応じて、４分木（ＱＴ）モードを用いて符号化ブロックを分割することと、
を装置に実行させるために、命令を実行するように構成される、装置。
２５．少なくとも１つのプロセッサが、
符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されることに応じて、２分木（ＢＴ）モード及び３分木（ＴＴ）モードは、符号化ブロックを分割するために用いられることを許可されないと決定すること
を装置に実行させるために、命令を実行するように構成される、条項２４に記載の装置。
２６．少なくとも１つのプロセッサが、
ＱＴフラグが符号化ブロックを含むビットストリーム内に存在するかどうかに関わらず、ＱＴモードを用いて符号化ブロックを分割することであって、ＱＴフラグが、符号化ブロックがＱＴモードを用いて分割されるかどうかを示すこと
を装置に実行させるために、命令を実行するように構成される、条項２４及び２５のいずれか一項に記載の装置。
２７．少なくとも１つのプロセッサが、
ＱＴモードの適用が許可される最小ブロックサイズに対するプリセット制約に関わらず、ＱＴモードを用いて符号化ブロックを分割すること
を装置に実行させるために、命令を実行するように構成される、条項２４に記載の装置。
２８．プリセット制約が、符号化ブロックに関連付けられたビットストリーム適合性を含み、ビットストリーム適合性が、ＱＴモードの適用が許可される最小ブロックサイズを設定する、条項２７に記載の装置。
２９．最小ブロックサイズが、６４以下であるように設定される、条項２８に記載の装置。
３０．ビットストリーム適合性が、最大ＢＴ深さ又は最大ＴＴ深さを設定する、条項２８及び２９のいずれか一項に記載の装置。
３１．ピクチャ境界が、下ピクチャ境界又は右ピクチャ境界である、条項２４～３０のいずれか一項に記載の装置。
３２．命令のセット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、命令のセットが、
符号化ブロックがピクチャ境界の外側にサンプルを含むかどうかを決定することと、
符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されることに応じて、４分木（ＱＴ）モードを用いて符号化ブロックを分割することと、
を映像処理装置に実行させるために、１つ以上の処理デバイスによって実行可能である、非一時的コンピュータ可読記憶媒体。
３３．命令のセットが、
符号化ブロックがピクチャ境界の外側にサンプルを含むと決定されることに応じて、２分木（ＢＴ）モード及び３分木（ＴＴ）モードは、符号化ブロックを分割するために用いられることを許可されないと決定すること
を映像処理装置に実行させるために、１つ以上の処理デバイスによって実行可能である、条項３２に記載の非一時的コンピュータ可読記憶媒体。
３４．命令のセットが、
ＱＴフラグが符号化ブロックを含むビットストリーム内に存在するかどうかに関わらず、ＱＴモードを用いて符号化ブロックを分割することであって、ＱＴフラグが、符号化ブロックがＱＴモードを用いて分割されるべきかどうかを示すこと
を映像処理装置に実行させるためにに、１つ以上の処理デバイスによって実行可能である、条項３２及び３３のいずれか一項に記載の非一時的コンピュータ可読記憶媒体。
３５．命令のセットが、
ＱＴモードの適用が許可される最小ブロックサイズに対するプリセット制約に関わらず、ＱＴモードを用いて符号化ブロックを分割すること
を映像処理装置に実行させるために、１つ以上の処理デバイスによって実行可能である

、条項３２に記載の非一時的コンピュータ可読記憶媒体。
３６．プリセット制約が、符号化ブロックに関連付けられたビットストリーム適合性を含み、ビットストリーム適合性が、ＱＴモードの適用が許可される最小ブロックサイズを設定する、条項３５に記載の非一時的コンピュータ可読記憶媒体。
３７．最小ブロックサイズが、６４以下であるように設定される、条項３６に記載の非一時的コンピュータ可読記憶媒体。
３８．ビットストリーム適合性が、最大ＢＴ深さ又は最大ＴＴ深さを設定する、条項３６及び３７のいずれか一項に記載の非一時的コンピュータ可読記憶媒体。
３９．ピクチャ境界が、下ピクチャ境界又は右ピクチャ境界である、条項３２～３８のいずれか一項に記載の非一時的コンピュータ可読記憶媒体。

【０１４７】
[00168] 実施形態によっては、また、命令を含む非一時的コンピュータ可読記憶媒体が提供され、命令はデバイス（本開示の符号器及び復号器など）によって、上述の方法を遂行するために実行され得る。一般的な形態の非一時的媒体としては、例えば、フロッピー（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、又は任意の他の磁気データ記憶媒体、ＣＤ－ＲＯＭ、任意の他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、ＦＬＡＳＨ（登録商標）－ＥＰＲＯＭ又は任意の他のフラッシュメモリ、ＮＶＲＡＭ、キャッシュ、レジスタ、任意の他のメモリチップ又はカートリッジ、並びにこれらのネットワーク化バージョンが挙げられる。デバイスは、１つ以上のプロセッサ（ＣＰＵ）、入力／出力インターフェース、ネットワークインターフェース、及び／又はメモリを含み得る。

【０１４８】
[00169] 「第１（first）」及び「第２（second）」などの本明細書における関係語は、単に、実体又は動作を別の実体又は動作と区別するために使用されるにすぎず、これらの実体又は動作の間のいかなる実際の関係又は順序も必要とせず、暗示もしないことに留意されたい。さらに、単語「備える（comprising）」、「有する（having）」、「包含する（containing）」、及び「含む（including）」、並びに他の同様の形式は、意味が同等であり、これらの単語のうちの任意のものに続く要素若しくは要素群は、このような要素若しくは要素群の限定列挙であることを意味されない、又は列挙された要素若しくは要素群のみに限定されることを意味されないという点で、オープンエンドなものであることを意図される。

【０１４９】
[00170] 本明細書において使用するとき、別途特に断りのない限り、用語「又は（or）」は、実行不可能な場合を除き、全ての可能な組み合わせを包含する。例えば、データベースがＡ又はＢを含み得ると述べられた場合には、このとき、別途特に断りのない限り、又は実行不可能でない限り、データベースは、Ａ、或いはＢ、或いはＡ及びＢを含み得る。第２の例として、データベースがＡ、Ｂ、又はＣを含み得ると述べられた場合には、このとき、別途特に断りのない限り、又は実行不可能でない限り、データベースは、Ａ、Ｂ、若しくはＣ、又はＡ及びＢ、Ａ及びＣ、若しくはＢ及びＣ、又はＡ及びＢ及びＣを含み得る。

【０１５０】
[00171] 上述の実施形態は、ハードウェア、又はソフトウェア（プログラムコード）、或いはハードウェア及びソフトウェアの組み合わせによって実施され得ることが理解される。ソフトウェアによって実施される場合には、それは上述のコンピュータ可読媒体内に記憶され得る。ソフトウェアは、プロセッサによって実行されたときに、本開示の方法を遂行することができる。本開示において説明される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、或いはハードウェア及びソフトウェアの組み合わせによって実施され得る。当業者は、上述のモジュール／ユニットのうちの複数のものを１つのモジュール／ユニットとして組み合わせてもよく、上述のモジュール／ユニットの

各々を複数のサブモジュール／サブユニットにさらに分割してもよいことも理解するであろう。

【０１５１】

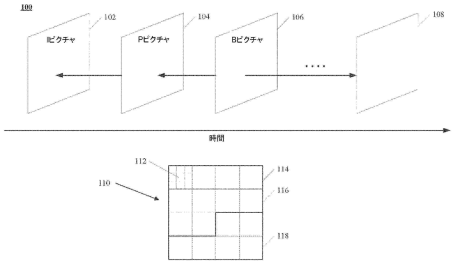
［00172］上述の明細書において、実施形態は、実装形態ごとに異なり得る数多くの特定の詳細を参照して説明された。上述の実施形態の特定の適応及び変更を行うことができる。本明細書の考慮及び本明細書において開示された本発明の実施から、他の実施形態が当業者に明らかになり得る。明細書及び実施例は例としてのみ考慮されることが意図されており、本発明の真の範囲及び趣旨は添付の請求項によって指示される。また、図に示されるステップの配列は単に例示目的のためのものにすぎず、ステップのいかなる特定の配列にも限定されることを意図されないことも意図される。それゆえ、当業者は、これらのステップは、同じ方法を実施しながらも、異なる順序で遂行され得ることを理解することができる。

【０１５２】

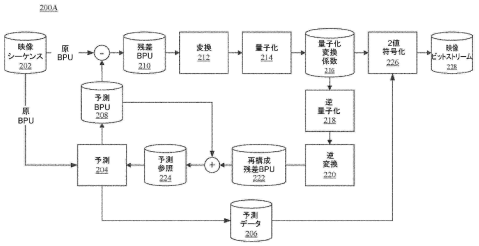
［00173］図面及び明細書において、例示的な実施形態が開示された。しかし、これらの実施形態に対して多くの変形及び変更を行うことができる。したがって、特定の用語が採用されていても、これらは単に、一般的な説明の意味で使用されているにすぎず、限定を目的として使用されているものではない。

【図面】

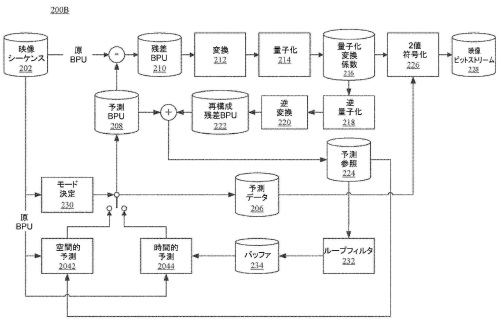
【図１】



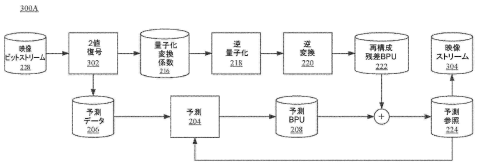
【図２Ａ】



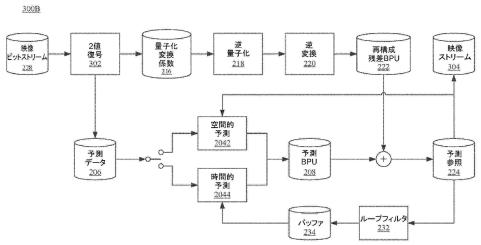
【図２Ｂ】



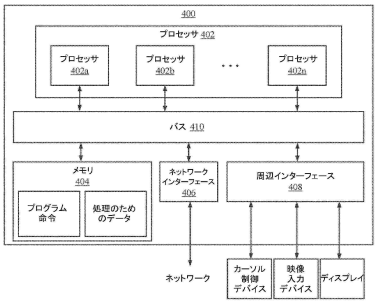
【図３Ａ】



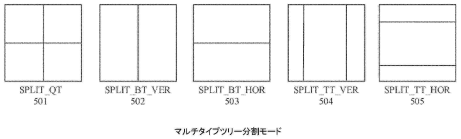
【図３Ｂ】



【図４】



【図５】



【図６】

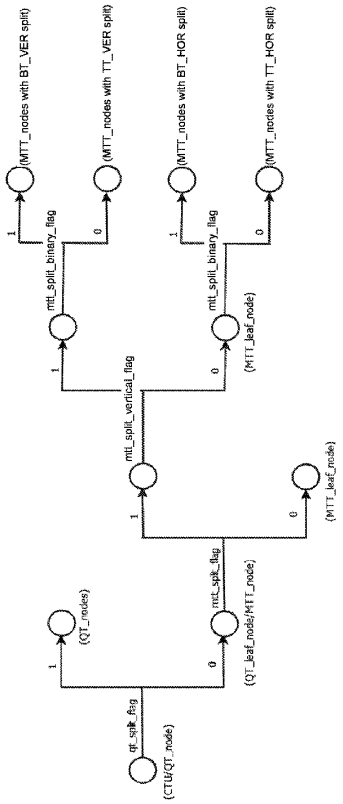


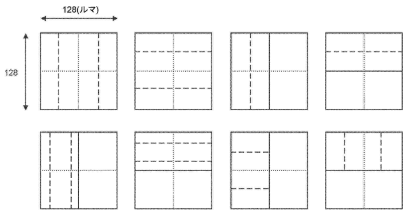
FIG. 6

【図７】

表1:マルチツリー構文要素に基づくMttSplitMode導出

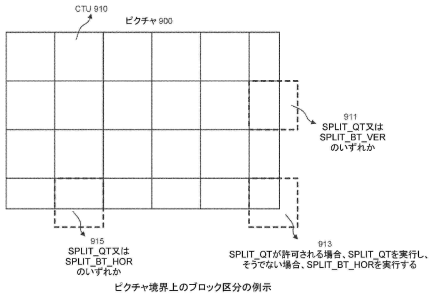
MttSplitMode	mtt_split_cu_vertical_flag	mtt_split_cu_binary_flag
SPLIT_TT_HOR	0	0
SPLIT_BT_HOR	0	1
SPLIT_TT_VER	1	0
SPLIT_BT_VER	1	1

【図８】



許可されないTT及びBT区分の例

【図９】



ピクチャ境界上のブロック区分の例示

【図１０】

表2:btSplitに基づくparallelTtSplit及びcbSizeの仕様

	btSplit == SPLIT_BT_VER	btSplit == SPLIT_BT_HOR
parallelTtSplit	SPLIT_TT_VER	SPLIT_TT_HOR
cbSize	cbWidth	cbHeight

【図１１】

表3:ttSplitに基づくcbSizeの仕様

	ttSplit == SPLIT_TT_VER	ttSplit == SPLIT_TT_HOR
cbSize	cbWidth	cbHeight

【図１２Ａ】

表4:7.3.9.4節　符号化ツリー構文<パートI>

	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 <= pic_width_in_luma_samples) && (y0 + cbHeight <= pic_height_in_luma_samples))	
1201	***split_cu_flag***	ae(v)
	if(cu_qp_delta_enabled_flag && qgOnY && cbSubdiv <= CuQpDeltaSubdiv) {	
	IsCuQpDeltaCoded = 0	
	CuQpDeltaVal = 0	
	CuQgTopLeftX = x0	
	CuQgTopLeftY = y0	
	}	
	if(cu_chroma_qp_offset_enabled_flag && qgOnC && cbSubdiv <= CuChromaQpOffsetSubdiv)	
	IsCuChromaQpOffsetCoded = 0	
	if(split_cu_flag) {	
	if((allowSplitBtVer || allowSplitBtHor || allowSplitTtVer || allowSplitTtHor) && allowSplitQT)	
1202	***split_qt_flag***	ae(v)
	if(!split_qt_flag) {	
	if((allowSplitBtHor || allowSplitTtHor) && (allowSplitBtVer || allowSplitTtVer))	
1203	***mtt_split_cu_vertical_flag***	ae(v)
	if((allowSplitBtVer && allowSplitTtVer && mtt_split_cu_vertical_flag) || (allowSplitBtHor && allowSplitTtHor && !mtt_split_cu_vertical_flag))	
1204	***mtt_split_cu_binary_flag***	ae(v)
	}	
	if(modeTypeCondition == 1)	
	modeType = MODE_TYPE_INTRA	
	else if(modeTypeCondition == 2) {	
	mode_constraint_flag	ae(v)
	modeType = mode_constraint_flag ? MODE_TYPE_INTRA : MODE_TYPE_INTER	
	} else {	
	modeType = modeTypeCurr	
	}	

【図１２Ｂ】

表4:7.3.9.4節　符号化ツリー構文<パートII>

treeType = (modeType == MODE_TYPE_INTRA) ? DUAL_TREE_LUMA : treeTypeCurr	
if(!split_qt_flag) {	
if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_VER) {	
depthOffset += (x0 + cbWidth > 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 < pic_width_in_luma_samples)	
coding_tree(x1, y0, cbWidth / 2, cbHeightY, qgOnY, qgOnC, cbSubdiv + 1, cqtDepth, mttDepth + 1, depthOffset, 1, treeType, modeType)	
} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_HOR) {	
depthOffset += (y0 + cbHeight > 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 < 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)	

(続き)

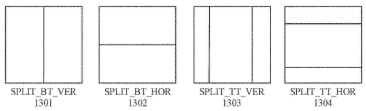
【図１２Ｃ】

表4:7.3.9.4節　符号化ツリー構文<パートIII>

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)	
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 < 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 < 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 < pic_height_in_luma_samples && x1 < 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(modeTypeCur == 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)	
}	

(続き)

【図１３】



MttSplitModeによって示されるマルチタイプツリー分割モード(参考)

【図１４】

表5:x=x0..x0+cbWidth-1且つy=y0..y0+cbHeight-1の間の、
MttSplitMode[x][y][mttDepth]の仕様

MttSplitMode[x0][y0][mttDepth]	mtt_split_cu_vertical_flag	mtt_split_cu_binary_flag
SPLIT_TT_HOR	0	0
SPLIT_BT_HOR	0	1
SPLIT_TT_VER	1	0
SPLIT_BT_VER	1	1

【図１５】

表6:7.3.2.3節 シーケンスパラメータセットRBSP構文

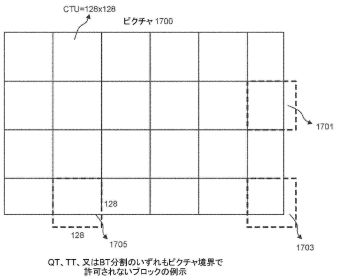
seq_parameter_set_rbsp( ) {	記述子
…	
if(ChromaArrayType != 0)	
qtbtt_dual_tree_intra_flag	u(1)
log2_min_luma_coding_block_size_minus2	ue(v)
partition_constraints_override_enabled_flag	u(1)
sps_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
sps_log2_diff_min_qt_min_cb_inter_slice	ue(v)
sps_max_mtt_hierarchy_depth_inter_slice	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_slice != 0) {	
sps_log2_diff_max_bt_min_qt_inter_slice	ue(v)
sps_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
if(qtbtt_dual_tree_intra_flag) {	
sps_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
…	
}	

【図１６】

表7:7.3.2.6節 ピクチャヘッダRBSP構文

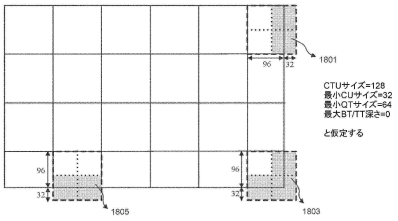
picture_header_rbsp( ) {	記述子
…	
if(partition_constraints_override_enabled_flag) {	
partition_constraints_override_flag	u(1)
if(partition_constmints_override_flag) {	
pic_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
pic_log2_diff_min_qt_min_cb_inter_slice	ue(v)
pic_max_mtt_hierarchy_depth_inter_slice	ue(v)
pic_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(pic_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
pic_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
pic_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
if(pic_max_mtt_hierarchy_depth_inter_slice != 0) {	
pic_log2_diff_max_bt_min_qt_inter_slice	ue(v)
}	
if(qtbtt_dual_tree_intra_flag) {	
pic_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
pic_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if(pic_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
pic_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
pic_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
}	
}	
…	
}	

【図１７】



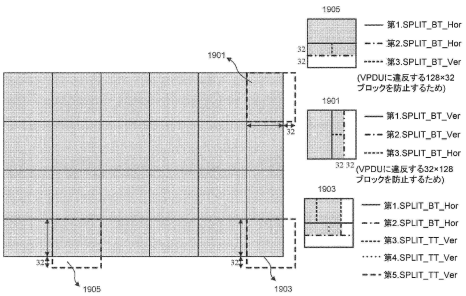
QT、TT、又はBT分割のいずれもピクチャ境界で
許可されないブロックの例示

【図１８】

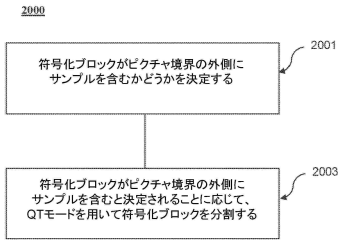


QT、TT、又はBT分割のいずれもピクチャ境界で
許可されないブロックの例示

【図１９】



【図２０】



フロントページの続き

(72)発明者　リャオ，ルーリン
中華人民共和国，ベイジン チャオヤン ディストリクト ザ フォース ディストリクト ワンジンドン ユァン ラディアンス ジン フイ タワー
(72)発明者　イエ，ヤン
アメリカ合衆国，カリフォルニア州 ９２１３０ サンディエゴ パールマン ウェイ ５００１
(72)発明者　チェン，ジエ
中華人民共和国，ベイジン チャオヤン ディストリクト ザ フォース ディストリクト ワンジンドン ユァン ビルディング ナンバー６
(72)発明者　ルオ，ジャンコン
アメリカ合衆国，ニューヨーク州 ０８５５８ スキルマン リッチモンド ドライブ ２８
審査官　田中 純一

(56)参考文献　特表２０２２－５２４４４１（ＪＰ，Ａ）
米国特許出願公開第２０２２／０１８２６２３（ＵＳ，Ａ１）
特表２０２０－５１６１５６（ＪＰ，Ａ）
国際公開第２０１８／１７７７４１（WO，Ａ１）

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