

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7547592号
(P7547592)

(45)発行日 令和6年9月9日(2024.9.9)

(24)登録日 令和6年8月30日(2024.8.30)

(51)国際特許分類		F I	
H O 4 N	19/119 (2014.01)	H O 4 N	19/119
H O 4 N	19/157 (2014.01)	H O 4 N	19/157
H O 4 N	19/176 (2014.01)	H O 4 N	19/176
H O 4 N	19/70 (2014.01)	H O 4 N	19/70
請求項の数 15 外国語出願 (全35頁)			
(21)出願番号	特願2023-170188(P2023-170188)	(73)特許権者	504161984
(22)出願日	令和5年9月29日(2023.9.29)		ホアウェイ・テクノロジーズ・カンパニ
(62)分割の表示	特願2020-566735(P2020-566735)		ー・リミテッド
原出願日	令和1年5月29日(2019.5.29)		中華人民共和国・518129・グァン
(65)公開番号	特開2023-179590(P2023-179590)		ドン・シェンツェン・ロンガン・ディス
	A)		トリクト・バンティアン・(番地なし)
(43)公開日	令和5年12月19日(2023.12.19)		・ホアウェイ・アドミニストレーション
審査請求日	令和5年10月27日(2023.10.27)	(74)代理人	・ビルディング
(31)優先権主張番号	62/678,241		100110364
(32)優先日	平成30年5月30日(2018.5.30)	(74)代理人	弁理士 実広 信哉
(33)優先権主張国・地域又は機関	米国(US)		100133569
		(72)発明者	弁理士 野村 進
			ハン・ガオ
			ドイツ・80992・ミュンヘン・リー
			スシュトラーセ・25・ホアウェイ・テ
			最終頁に続く

(54)【発明の名称】 ピクチャ境界処理のためのマルチタイプツリー深度拡張

(57)【特許請求の範囲】

【請求項1】

画像を符号化ユニットに分割するための方法であって、
水平方向および垂直方向においてサイズを有する非境界符号化ツリーユニット（CTU）と、水平または垂直の画像境界によって区切られた前記画像内の部分を有する境界CTUとを含むCTUに前記画像を細分化するステップであって、前記部分は、前記画像境界に垂直な方向において前記サイズより小さいサイズを有する、ステップと、
前記非境界CTUおよび前記境界CTUをそれぞれの符号化ユニットに階層的に区分化するステップと
を含み、
前記非境界CTUの前記階層的な区分化は、非境界マルチタイプ区分最大深度を有するマルチタイプ分割を含み、マルチタイプ分割は、分割方向が前記垂直方向または前記水平方向のいずれかである分割であり、
前記境界CTUの前記階層的な区分化は、境界マルチタイプ区分最大深度を有するマルチタイプ分割を含み、
 $Ext\max BTTDepth = BTTBDepth + \max BTTDepth$ であり、
 $Ext\max BTTDepth$ は前記境界マルチタイプ区分最大深度を示し、
 $BTTBDepth$ は適応境界マルチタイプ区分深度を示し、
 $\max BTTDepth$ は事前定義されたマルチタイプ区分深度を示す、画像を分割するための方法。

【請求項 2】

前記適応境界マルチタイプ区分深度は、二分木（BT）境界区分化（BP）の1つの層が存在する場合、1だけ増加される、請求項1に記載の、画像を分割するための方法。

【請求項 3】

前記適応境界マルチタイプ区分深度は、分割方向が前記画像境界に垂直である場合、増加されない、請求項1または2に記載の、画像を分割するための方法。

【請求項 4】

前記適応境界マルチタイプ区分深度は、二分木および三分木（BTT）境界区分化（BP）のN個の層が存在する場合、Nだけ増加される、請求項1から3のいずれか一項に記載の、画像を分割するための方法。

【請求項 5】

Nは1、2、3、または4に等しい、請求項4に記載の、画像を分割するための方法。

【請求項 6】

ビデオシーケンスの画像をエンコードするための方法であって、
請求項1から5のいずれか一項に従って画像を符号化ユニットに分割するステップと、
前記符号化ユニットをエンコードする画像符号化ステップと、
前記エンコードされた符号化ユニット、および前記符号化ツリーユニットがどのように区分化されたかを示す区分化情報を含むビットストリームを生成するビットストリーム形成ステップと
を含む、方法。

【請求項 7】

ビデオシーケンスの画像をデコードするための方法であって、
エンコードされた符号化ユニットを含むビットストリームを解析するステップと、
請求項1から5のいずれか一項に従って画像の分割を決定する前記ステップと、
前記画像の前記決定された分割に基づいて、前記エンコードされた符号化ユニットをデコードする画像デコーディングステップと
を含む、方法。

【請求項 8】

画像を符号化ユニットに分割するための装置であって、前記装置は処理ユニットを含み、前記処理ユニットは、
水平方向および垂直方向においてサイズを有する非境界符号化ツリーユニット（CTU）と、水平または垂直の画像境界によって区切られた前記画像内の部分を有する境界CTUとを含むCTUに前記画像を細分化するように構成され、前記部分は、前記画像境界に垂直な方向において前記サイズより小さいサイズを有し、
前記処理ユニットは、前記非境界CTUおよび前記境界CTUをそれぞれの符号化ユニットに階層的に区分化するように構成され、
前記非境界CTUの前記階層的な区分化は、非境界マルチタイプ区分最大深度を有するマルチタイプ分割を含み、マルチタイプ分割は、分割方向が前記垂直方向または前記水平方向のいずれかである分割であり、
前記境界CTUの前記階層的な区分化は、境界マルチタイプ区分最大深度を有するマルチタイプ分割を含み、
$$\text{ExtMaxBTTDepth} = \text{BTTBPDepth} + \text{MaxBTTDepth}$$
であり、
ExtMaxBTTDepthは前記境界マルチタイプ区分最大深度を示し、
BTTBPDepthは適応境界マルチタイプ区分深度を示し、
MaxBTTDepthは事前定義されたマルチタイプ区分深度を示す、画像を分割するための装置。

【請求項 9】

前記適応境界マルチタイプ区分深度は、二分木（BT）境界区分化（BP）の1つの層が存在する場合、1だけ増加される、請求項8に記載の、画像を分割するための装置。

【請求項 10】

10

20

30

40

50

前記適応境界マルチタイプ区分深度は、分割方向が前記画像境界に垂直である場合、増加されない、請求項8または9に記載の、画像を分割するための装置。

【請求項11】

前記適応境界マルチタイプ区分深度は、二分木および三分木（BTT）境界区分化（BP）のN個の層が存在する場合、Nだけ増加される、請求項8から10のいずれか一項に記載の、画像を分割するための装置。

【請求項12】

Nは1、2、3、または4に等しい、請求項11に記載の、画像を分割するための装置。

【請求項13】

ビデオシーケンスの画像をエンコードするための装置であって、
請求項8から12のいずれか一項に記載の、画像を分割するための装置と、
前記符号化ユニットをエンコードするように構成された画像符号化ユニットと、
前記エンコードされた符号化ユニット、および前記符号化ツリーユニットがどのように区分化されたかを示す区分化情報を含むビットストリームを生成するように構成されたビットストリーム形成ユニットと
を含む、装置。

10

【請求項14】

ビデオシーケンスの画像をデコードするための装置であって、
エンコードされた符号化ユニットを含むビットストリームを解析するためのビットストリームパーサと、
請求項8から12のいずれか一項に従って画像を符号化ユニットに分割するための装置と、
前記画像の決定された前記分割に基づいて、前記エンコードされた符号化ユニットをデコードするための画像デコーディングユニットと
を含む、装置。

20

【請求項15】

処理回路によって実行されると、前記処理回路に請求項1から6のいずれか一項に記載の方法を実行させる命令を記憶したコンピュータ可読媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、ビデオ処理の分野、特に、ハイブリッドビデオ符号化および圧縮と通常呼ばれるトピックに関する。

30

【背景技術】

【0002】

Versatile Video Coding (VVC) 次世代規格は、Joint Video Exploration Team (JVET) として知られるパートナーシップで協力するITU-T Video Coding Experts Group (VCEG) とISO/IEC Moving Picture Experts Group (MPEG) の標準化団体の最新の共同ビデオプロジェクトである。

【0003】

現在のブロックベースのハイブリッドビデオコーデックは、予測符号化を用いている。ビデオシーケンスのピクチャは、ピクセルのブロックに細分化され、次にこれらのブロックは符号化される。ブロックをピクセルごとに符号化する代わりに、ブロック全体が、ブロックに空間的または時間的に近接する既にエンコードされたピクセルを使用して予測される。エンコーダは、ブロックとその予測との差のみをさらに処理する。さらなる処理は、通常、ブロックピクセルの、変換領域内の係数への変換を含む。次に、ビットストリームを形成するために、係数は、さらに圧縮され（例えば、量子化によって）、さらにコンパクト化され得る（例えば、エントロピー符号化によって）。ビットストリームは、デコーダがエンコードされたビデオをデコードすることを可能にする任意のシグナリング情報をさらに含み得る。例えば、シグナリングは、入力ピクチャのサイズ、フレームレート、量子化ステップ指示、またはピクチャのブロックに適用される予測などのエンコーダ設定

40

50

に関する設定を含み得る。

【0004】

ブロックとその予測との差は、ブロックの残差として知られている。より具体的には、ブロックの各ピクセルは、そのピクセルの強度レベルとその予測強度レベルとの差である残差を有する。ピクセルの強度レベルは、ピクセル値またはピクセルの値と呼ばれる。ブロックのすべてのピクセルの残差は、まとめてブロックの残差と呼ばれる。言い換えれば、ブロックは、ブロックのすべてのピクセルの残差からなるセットまたは行列である残差を有する。次に、残差は変換され、量子化され、シグナリング情報とともに符号化される。符号化は、算術符号化または他のエントロピー符号化タイプを含む様々な形態の固定長および可変長符号化を含み得る。

10

【0005】

ブロックベースのハイブリッドビデオ符号化では、各ピクチャがサンプルのブロックに区分化され、独立してデコード可能なエンティティとしてスライスを形成するために、ピクチャ内の複数のブロックが集約される。予測および／または変換が適用されるブロックは、符号化ユニット（CU）または符号化ブロック（CB）と呼ばれる。符号化ユニットは異なるサイズを有し得る。

【0006】

例えば、高効率ビデオ符号化（HEVC、H. 265としても知られている）では、ビデオフレームは、符号化ツリーユニット（CTU、符号化ツリーブロック、CTBとも呼ばれる）に細分化される。CTBは、同じサイズの互いに素な正方形のブロック、例えば、64×64の

20

サンプルである。各CTBは、ブロック区分化四分木構造、符号化ツリーのルートとして機能する。CTBは、符号化ツリー構造に沿って符号化ブロックにさらに細分化され得る。符号化ブロックに対して、予測タイプが決定される。符号化ブロックは、変換および量子化が適用されるより小さな変換ブロックにさらに分割され得る。

【0007】

HEVCでの区分化に関する詳細は、V. Sze et al (Ed.), High Efficiency Video Coding (HEVC) : Algorithms and Architectures, Springer, 2014, Chapter 3. 2に見出され得る。

【0008】

加えて、国際公開第2016／090568号は、四分木と二分木構造を使用して、ユニットを複数のより小さなユニットに区分化するための二分木構造を示している。したがって、ルートユニットは、最初に四分木構造によって区分化され、次に四分木のリーフノードは、二分木構造によってさらに区分化される。

30

【先行技術文献】

【特許文献】

【0009】

【文献】国際公開第2016／090568号

【非特許文献】

【0010】

【文献】V. Sze et al (Ed.), High Efficiency Video Coding (HEVC) : Algorithms and Architectures, Springer, 2014, Chapter 3. 2

40

【発明の概要】

【課題を解決するための手段】

【0011】

本発明の実施形態は、独立請求項の特徴によって規定され、実施形態のさらに好適な実施形態は、従属請求項の特徴によって規定される。

【0012】

一般的な態様によれば、本開示は、画像を符号化ユニットに分割するための装置を提供し、本装置は、処理回路を含む。本装置は、符号化ツリーユニットCTUであって、水平方向および垂直方向の所定のサイズを有する非境界CTUと、水平または垂直の画像境界によ

50

って区切られた画像内の部分を有する境界CTUであって、部分は、画像境界に垂直な方向に所定のサイズより小さいサイズを有する、境界CTUを含む符号化ツリーユニットCTUに画像を細分化し、非境界CTUおよび境界CTUをそれぞれの符号化ユニットに階層的に区分化し、非境界CTUの階層的区分化は、非境界マルチタイプ区分最大深度を有するマルチタイプ分割であって、分割方向が垂直方向または水平方向のいずれかであるマルチタイプ分割を含み、境界CTUの階層的区分化は、境界マルチタイプ区分最大深度を有するマルチタイプ分割を含む、ように構成されている。

【0013】

これは、境界区分化の柔軟性が高められるという利点を提供する。

【0014】

本装置のさらなる実施態様では、境界マルチタイプ区分最大深度は、少なくとも適応境界マルチタイプ区分深度と、事前定義されたマルチタイプ区分深度との合計であり、適応境界マルチタイプ区分深度は、分割方向が画像境界の方向であるマルチタイプ分割の深度である。

【0015】

これは、境界符号化ツリーユニットまたは区分ブロックにマルチタイプ分割を使用するときの区分深度の適応的決定を実現する。

【0016】

例えば、事前定義されたマルチタイプ区分深度は、非境界マルチタイプ区分最大深度に等しい。

【0017】

これは、非境界マルチタイプ区分最大深度を再使用することを実現する。

【0018】

本装置のさらなる実施態様では、合計は、境界CTUの境界区分ブロックの画像境界の方向および画像境界に垂直な方向のサイズの比の関数をさらに含み、境界区分ブロックは、適応境界マルチタイプ区分深度のブロックである。

【0019】

これは、マルチタイプ境界区分化の最大深度をさらに増大させ、これにより区分化の柔軟性を高めることを実現する。

【0020】

例えば、関数は2進対数である。

【0021】

これは、実用的な実施態様を実現するため、有益である。

【0022】

いくつかのさらなる実施形態では、境界マルチタイプ区分最大深度が事前定義される。

【0023】

これは、階層的区分化を決定する際の計算コストを低減することを容易にする。

【0024】

例えば、境界CTUの階層分割は、四分木分割をさらに含む。

【0025】

これは、さまざまなモードからの柔軟な選択を実現する。

【0026】

本装置のさらなる実施態様では、境界マルチタイプ区分最大深度は、非境界マルチタイプ区分最大深度以上である。

【0027】

これは、可能な境界区分化最大深度を高めることを実現する。

【0028】

上記の例および実施形態のいずれかによる、画像を符号化ユニットに分割するための装置を備える、ビデオシーケンスの画像をエンコードするための装置がさらに提供される。本装置は、符号化ユニットをエンコードするように構成された画像符号化ユニットと、エ

10

20

30

40

50

ンコードされた符号化ユニット、および符号化ツリーユニットがどのように区分化されたかを示す区分化情報を含むビットストリームを生成するように構成されたビットストリーム形成ユニットとをさらに備える。

【0029】

さらなる実施態様では、画像をエンコードするための装置は、画像を分割するための装置を含み、境界マルチタイプ区分最大深度は事前定義され、ビットストリームは、境界マルチタイプ区分化最大深度を含むエンコードされたシーケンスパラメータセットをさらに含む。

【0030】

エンコードされた符号化ユニットを含むビットストリームを解析するためのビットストリームパーサと、上記の例および実施形態のいずれかによる、画像の分割を決定するための装置と、画像の決定された分割に基づいて、エンコードされた符号化ユニットをデコードするための画像デコーディングユニットとを備える、ビデオシーケンスの画像をデコードするための装置がさらに提供される。

10

【0031】

さらなる実施態様では、画像をデコードするための装置は、画像の分割を決定するための装置を含み、境界マルチタイプ区分最大深度は事前定義され、ビットストリームは、境界マルチタイプ区分化最大深度を含むエンコードされたシーケンスパラメータセットをさらに含み、画像の分割を決定するための装置は、シーケンスパラメータセットから第2のマルチタイプ区分化最大深度を取得するようにさらに構成されている。

20

【0032】

別の一般的な態様によれば、画像を符号化ユニットに分割するための方法が提供される。本方法は、符号化ツリーユニットCTUであって、水平方向および垂直方向の所定のサイズを有する非境界CTUと、水平または垂直の画像境界によって区切られた画像内の部分を有する境界CTUであって、部分は、画像境界に垂直な方向に所定のサイズより小さいサイズを有する、境界CTUを含む符号化ツリーユニットCTUに画像を細分化するステップと、非境界CTUおよび境界CTUをそれぞれの符号化ユニットに階層的に区分化するステップであって、非境界CTUの階層的区分化は、非境界マルチタイプ区分最大深度を有するマルチタイプ分割であって、分割方向が垂直方向または水平方向のいずれかであるマルチタイプ分割を含み、境界CTUの階層的区分化は、境界マルチタイプ区分最大深度を有するマルチタイプ分割を含む、ステップとを含む。

30

【0033】

本方法のさらなる実施態様では、境界マルチタイプ区分最大深度は、少なくとも適応境界マルチタイプ区分深度と、事前定義されたマルチタイプ区分深度との合計であり、適応境界マルチタイプ区分深度は、分割方向が画像境界の方向であるマルチタイプ分割の深度である。

【0034】

例えば、事前定義されたマルチタイプ区分深度は、非境界マルチタイプ区分最大深度に等しい。

【0035】

本方法のさらなる実施態様では、合計は、境界CTUの境界区分ブロックの画像境界の方向および画像境界に垂直な方向のサイズの比の関数をさらに含み、境界区分ブロックは、適応境界マルチタイプ区分深度のブロックである。

40

【0036】

例えば、関数は2進対数である。

【0037】

別の実施形態では、境界マルチタイプ区分最大深度が事前定義される。

【0038】

さらなる実施態様では、境界CTUの階層分割は、四分木分割をさらに含む。

【0039】

50

例えば、境界マルチタイプ区分最大深度は、非境界マルチタイプ区分最大深度以上である。

【0040】

ビデオシーケンスの画像をエンコードするための方法であって、該方法は、上記の実施形態のいずれかに従って画像を符号化ユニットに分割するステップと、符号化ユニットをエンコードする画像符号化ステップと、エンコードされた符号化ユニット、および符号化ツリーユニットがどのように区分化されたかを示す区分化情報を含むビットストリームを生成するビットストリーム形成ステップとを含む、方法がさらに提供される。

【0041】

さらなる実施態様では、画像をエンコードするための方法は、画像を分割するための方法を含み、境界マルチタイプ区分最大深度は事前定義され、ビットストリームは、境界マルチタイプ区分最大深度を含むエンコードされたシーケンスパラメータセットをさらに含む。

【0042】

ビデオシーケンスの画像をデコードするための方法であって、該方法は、エンコードされた符号化ユニットを含むビットストリームを解析するステップと、上記の実施形態のいずれかに従って画像の分割を決定するステップと、画像の決定された分割に基づいて、エンコードされた符号化ユニットをデコードする画像デコーディングステップとを含む、方法がさらに提供される。

【0043】

さらなる実施態様では、画像をデコードするための方法は、画像の分割を決定するための方法を含み、境界マルチタイプ区分最大深度は事前定義され、ビットストリームは、境界マルチタイプ区分最大深度を含むエンコードされたシーケンスパラメータセットをさらに含み、画像の分割を決定するための方法は、シーケンスパラメータセットから第2のマルチタイプ区分最大深度を取得するステップをさらに含む。

【0044】

さらなる態様として、本開示は、処理回路によって実行されると、処理回路に、画像を符号化ユニットに分割するための方法、ビデオシーケンスの画像をエンコードするための方法、または上記の実施形態のいずれかに従ってビデオシーケンスの画像をデコードするための方法を実行させる命令を記憶したコンピュータ可読媒体を提供する。

【0045】

1つ以上の実施形態の詳細は、添付の図面および以下の説明に記載されている。他の特徴、目的、および利点は、説明、図面、および特許請求の範囲から明らかになる。

【0046】

本発明の以下の実施形態は、添付の図および図面を参照してより詳細に説明される。

【図面の簡単な説明】

【0047】

【図1】本発明の実施形態を実施するように構成されたビデオエンコーダの例示的な構造を示すブロック図である。

【図2】本発明の実施形態を実施するように構成されたビデオデコーダの例示的な構造を示すブロック図である。

【図3】HEVCによって用いられている四分木区分化の例を示す概略図である。

【図4】符号化ユニットを区分化するいくつかのモードを示す図である。

【図5】四分木／二分木の区分化の例を示す概略図である。

【図6】境界部分の強制四分木分割を示す概略図である。

【図7】境界部分の二分木分割を示す概略図である。

【図8】画像を符号化ユニットに分割するための装置のブロック図である。

【図9】境界部分の分割を示す概略図である。

【図10】二分木分割を使用する境界区分化の例を示す図である。

【図11】二分木分割を使用する境界区分化のさらなる例を示す図である。

10

20

30

40

50

【図12】四分木分割および二分木分割を使用した境界区分化の比較を示す図である。

【図13】マルチタイプツリー最大深度拡張の実施態様を示すフローチャートである。

【図14】本発明の実施形態を実施するように構成されたビデオ符号化システムの例を示すブロック図である。

【図15】ビデオ符号化デバイスの概略図である。

【発明を実施するための形態】

【0048】

本発明は、さらなる処理のために画像をより小さなユニットに分割（すなわち、区分化）することに関する。このような分割は、静止画像またはビデオ画像の符号化およびデコーディングで好適に使用され得る。以下では、本開示による分割を実施し得る例示的なビデオコーダおよびデコーダが説明される。

【0049】

図1は、ビデオストリームのフレームまたはピクチャの入力ブロックを受信するための入力と、エンコードされたビデオビットストリームを提供するための出力とを備えるエンコーダ100を示している。本開示における「フレーム」という用語は、ピクチャの同義語として使用される。しかしながら、本開示は、インターレースが適用される場合の分野にも適用可能であることに留意されたい。一般に、ピクチャは $m \times n$ のピクセルを含む。これらは、画像サンプルに対応し、それぞれ1つ以上の色成分を含み得る。簡単にするために、以下の説明は、輝度のサンプルを意味するピクセルに言及する。しかしながら、本開示の分割アプローチは、クロミナンスを含む任意の色成分またはRGBなどの色空間の成分に適用され得ることに留意されたい。一方、1つの成分のみに対して分割を実行し、決定された分割をより多くの（またはすべての）残りの成分に適用することが有益な場合がある。

【0050】

エンコーダ100は、区分化、予測、変換、量子化、およびエントロピー符号化をビデオストリームに適用するように構成される。

【0051】

分割ユニット110では、入力ビデオフレームは、符号化前にさらに分割される。符号化されるブロックは、必ずしも同じサイズを有さない。1つのピクチャは、異なるサイズのブロックを含む場合があり、ビデオシーケンスの異なるピクチャのブロックラストも異なる場合がある。特に、各ビデオ画像（ピクチャ）は、最初に同じ固定サイズのCTUに細分化される。CTUサイズは、例えば標準で、固定および事前定義されてもよい。HEVCでは、 64×64 のサイズが使用される。しかしながら、本開示は、標準の固定サイズに限定されない。エンコーダで設定され、ビットストリーム内のシグナリングパラメータとして提供され得るCTUサイズを提供することが好適であり得る。例えば、異なるCTUサイズは、それぞれの異なるピクチャサイズおよび／またはコンテンツタイプにとって有益であり得る。CTUサイズは、任意のシグナリングレベルでシグナリングされてもよく、例えば、それは、ビデオシーケンス全体またはその部分（すなわち、複数のピクチャ）または個々のピクチャごとに共通であってもよい。これに対応して、それは、例えば、現在のコーデック（H.264/AVC、H.265/HEVC）から知られているピクチャパラメータセットPPS内もしくはシーケンスパラメータセットSPS内もしくはビデオパラメータセットVPS内または同様のパラメータセットでシグナリングされてもよい。代替的に、それは、スライスヘッダまたは任意の他のレベルで指定されてもよい。CTUサイズは、 64×64 とは異なる値をとる場合がある。それは、例えば 128×128 のサンプルの大きさの場合がある。一般に、二分木または四分木による階層分割を実行するには、2の累乗である、すなわち、 n が2より大きい整数である 2^n の形式であるCTUサイズを提供することが有益であり得る。

【0052】

ピクチャの、CTUへの区分化およびCTUの、CUへの区分化は、V. Sze et al (Ed.) , High Efficiency Video Coding (HEVC) : Algorithms and Architectures, Springer, 2014からの図3に示されている。区分化は、様々なローカル特性に適応するために四分木構造に従う。左側で図3は、右側の四分木構造に準拠して階層的に分割されたCT

10

20

30

40

50

Uを示している。特に、符号化ツリーは、CTUの、CUへの細分化を指定する構文を定義する。CTUと同様に、CUは、サンプルの正方形のブロックおよびこれらのサンプルブロックに関連付けられた構文からなる。したがって、区分化は、階層深度1の4つの（四分木内の）CUに細分化され得るが、細分化されなければならないわけではないCTU（階層深度0）から開始して階層的に実行される。図3では、CTUは、さらに分割されず、したがって四分木のリーフを形成する第1階層深度（レベル）のCU8および16と、階層深度2のCU（深度2のCU）にさらに分割される2つのさらなるCUとに分割されている。特に、左上の深度1のCUは、四分木リーフを形成する深度2のCU1、2、7と、すべてリーフである深度3のCU3、4、5、および6にさらに分割されているもう1つのCUとにさらに細分化されている。同様に、左下の深度1のCUは、四分木のリーフでもある深度2のCU13、14、および15と、すべてリーフであり、したがってそれ以上分割されないレベル3のCU9、10、11、および12にさらに分割されている残りのCUとにさらに分割されている。

10

【0053】

HEVCにおける四分木分割の例示的な構文は、以下で表1に示されている。

【0054】

【表1】

coding_quadtree(x0, y0, log2CbSize, cqtDepth) {	記述子
if(x0 + (1 << log2CbSize) <= pic_width_in_luma_samples && y0 + (1 << log2CbSize) <= pic_height_in_luma_samples && log2CbSize > MinCbLog2SizeY)	
split_cu_flag[x0][y0]	ae(v)
if(cu_qp_delta_enabled_flag && log2CbSize >= Log2MinCuQpDeltaSize) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
}	
if(cu_chroma_qp_offset_enabled_flag && log2CbSize >= Log2MinCuChromaQpOffsetSize)	
IsCuChromaQpOffsetCoded = 0	
if(split_cu_flag[x0][y0]) {	
x1 = x0 + (1 << (log2CbSize - 1))	
y1 = y0 + (1 << (log2CbSize - 1))	
coding_quadtree(x0, y0, log2CbSize - 1, cqtDepth + 1)	
if(x1 < pic_width_in_luma_samples)	
coding_quadtree(x1, y0, log2CbSize - 1, cqtDepth + 1)	
if(y1 < pic_height_in_luma_samples)	
coding_quadtree(x0, y1, log2CbSize - 1, cqtDepth + 1)	
if(x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples)	
coding_quadtree(x1, y1, log2CbSize - 1, cqtDepth + 1)	
} else	
coding_unit(x0, y0, log2CbSize)	
}	

20

30

40

表1：CTU 分割フラグの構文

【0055】

特に、CTUレベルでは、split_cu_flagと名付けられたフラグが、ビットストリームに含まれ、これは、完全なCTUがCUを形成するかどうか、またはそれが、正方形のサンプルブロックに対応する4つの等しいサイズのブロックに分割されるかどうかを示す。CTUが分割される場合、結果のブロックごとに、ブロックがCUを表すかどうか、またはそれが4

50

つの等しいサイズのブロックにさらに分割されるかどうかを指定する別の`split_cu_flag`が送信される。この階層的な細分化は、結果のブロックがさらに細分化されなくなるまで続けられる。CUの最小サイズは、シーケンスパラメータセットでシグナリングされ、それは、 8×8 のルマサンプル以上CTUのサイズ以内の範囲であり得る。階層的な細分化のプロセスで最小CUサイズが達せられると、対応するブロックに関して分割フラグは送信されず、代わりに、これらのブロックはそれ以上分割されないと推測される。通常のHEVCエンコーダ設定では、 8×8 から 64×64 のサンプルの範囲のCUが使用され得るように、サポートされているCUサイズの最大範囲が利用される。CTU内部のCUは、深度優先の順序で符号化される。この符号化順序は、zスキャンとも呼ばれる。それは、スライスの上または左の境界に位置するものを除く各CUについて、CUの上およびCUの左のすべてのサンプルが既に符号化されており、このため、対応するサンプルが、イントラ予測に使用され得、関連する符号化パラメータが、現在のCUの符号化パラメータを予測するために使用され得ることを保証する。

【0056】

言い換えれば、`split_cu_flag[x0][y0]`は、符号化ユニットが水平方向および垂直方向の半分の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。配列インデックス $x0$, $y0$ は、ピクチャの左上のルマサンプルに対する、当該の符号化ブロックの左上のルマサンプルの位置($x0$, $y0$)を指定する。`split_cu_flag[x0][y0]`が存在しない場合、以下がデコーダに当てはまり、

`log2CbSize`(符号化ブロックサイズを指定するパラメータ)が`MinCbLog2SizeY`(構成可能な最小符号化ユニットサイズを指定するパラメータ)より大きい場合、`split_cu_flag[x0][y0]`の値は1に等しいと推測される。

そうでない場合(`log2CbSize`が`MinCbLog2SizeY`に等しい)、`split_cu_flag[x0][y0]`の値は0に等しいと推測される。

【0057】

配列`CtDepth[x][y]`は、位置(x , y)をカバーするルマ符号化ブロックの符号化ツリー深度を指定する。`split_cu_flag[x0][y0]`が0に等しい場合、`CtDepth[x][y]`は、 $x=x0 \dots x0+nCbS-1$ および $y=y0 \dots y0+nCbS-1$ に関して`cqtDepth`に等しいと推測される。

【0058】

VVC(多用途ビデオ符号化)では、四分木(QT)セグメント化およびマルチタイプ(二分/三分/非対称二分タイプ)ツリー(BT/TT/ABT)セグメント化構造を含むセグメント化構造が、複数の区分ユニットタイプの概念に取って代わる。言い換えれば、新しいセグメント化構造は、最大変換長に対してあまりに大きいサイズを有するCUに必要な場合を除いて、CU(符号化ユニット)、PU(予測ユニット)、およびTU(変換ユニット)の概念の分離を除去し、CU区分形状に関してより高い柔軟性をサポートする[JVET-J1002]。図4は、VTM(VVCテストモデル)で現在使用されている区分モードを示している。図4のパート(a)は、それ以上分割が適用されない(分割なしの)CTUまたはCUを示している。パート(b)は、CTUまたはCUが垂直方向と水平方向の両方で分割されるクォータナリツリー(一般的には「四分木」とも呼ばれる)分割モードを示している。パート(c)および(d)は、それぞれ垂直方向および水平方向の二分木分割モードを示している。さらに、パート(e)および(f)は、垂直方向および水平方向の三分木分割を示している。三分木分割では、サイズ $1/4$ の2つのブロックおよびサイズ $1/2$ の1つのブロックがあることが理解され得る。

【0059】

CTUの階層的区分化に関連する以下のパラメータ、すなわち、

CTUサイズ: クォータナリツリーのルートノードサイズ

MinQTSize: 許可された最小のクォータナリツリーリーフノードサイズ

MaxBTSize: 許可された最大の二分木および三分木ルートノードサイズ

MaxBTDepth: 許可された二分木および三分木最大深度

MinBTTSize：許可された最小の二分木および三分木リーフノードサイズ

MinCUsSize：許可された最小のCUサイズ

は、BT/TT/QT符号化ツリースキームのシーケンスパラメータセット（SPS）の構文要素によって定義および指定される。

【0060】

図5は、四分木および二分木の混合区分化を示している。四分木区分化は実線で示され、一方、二分木区分化は破線で示されている。二分木によってさらに分割される符号化ユニットを表す、ノード上のラベル1または0は、それぞれ二分分割が垂直方向と水平方向のどちらで施されているかを示している。

【0061】

ルマサンプルにおけるビデオピクチャの水平および垂直サイズは、シーケンスパラメータセットで送信される、ルマサンプルにおける最小CUサイズの整数倍である必要があるが、それは、CTUサイズの整数倍である必要はない。ビデオピクチャの水平または垂直サイズがCTUサイズの整数倍を表していない場合、境界のCTUは、結果のブロックの境界がピクチャの境界と一致するまで分割されると推測される。この強制分割に関して、分割フラグは送信されないが、結果のブロックは、上で説明された四分木構文を使用してさらに分割され得る。ピクチャ領域の外部にあるCUは符号化されない。

【0062】

この分割は図6に示されており、これは、HDシーケンス（1920×1080）の下部境界のCTU（128×128）強制QT区分の図である。特に、図6は、フレーム境界であって、これより上の56本の線（128サンプルの長さ）がスライスまたは画像の境界部分である、フレーム境界を示している。フレーム境界より下のCTUの部分は、別のスライスに属する場合もあるし、または例えばフレーム境界がピクチャの下部境界である場合、まったく存在しない場合もある。理解され得るように、強制四分木分割が128×56のサンプルに適用されている。

【0063】

クロマCTBの細分化は、HEVCでは、それぞれのルマCTBの細分化と常に合わされる。本開示は、クロマ成分を同じように処理し得るが、それに限定されないことに留意されたい。異なる色成分の独立した分割であってもよい。

【0064】

図1に戻ると、分割ユニット110で画像分割を実行した後、エンコードされたビデオビットストリームとして出力を生成するために、変換ユニット130、量子化ユニット140、およびエントロピーエンコーディングユニット150によって変換、量子化、およびエントロピー符号化がそれぞれ実行される。

【0065】

ビデオストリームは、複数のフレームを含み得る。例えば、ビデオストリームの第1のフレームのブロックは、イントラ予測ユニット190によってイントラ符号化される。イントラフレームは、そのフレームのみからの情報を使用して符号化され、このため、それは、他のフレームから独立してデコードされ得る。したがって、イントラフレームは、例えばランダムアクセスのために、ビットストリームのエントリポイントを提供し得る。ビデオストリームの他のフレームのブロックは、インター予測ユニット195によってインター符号化され得、インター符号化されたフレームの各ブロックは、別のフレーム（参照フレーム）、例えば、以前に符号化されたフレーム内のブロックから予測される。モード選択ユニット180は、フレームのブロックにイントラ予測とインター予測のどちらが行われるか、すなわち、それがイントラ予測ユニット190とインター予測ユニット195のどちらによって処理されるかを選択するように構成される。モード選択ユニット180はまた、イントラまたはインター予測のパラメータを制御する。画像情報のリフレッシュを可能にするために、インター符号化されたフレームは、インター符号化されたブロックだけでなく、1つ以上のイントラ符号化されたブロックも含み得る。対照的に、イントラフレームは、イントラ符号化されたブロックのみを含み、インター符号化されたブロックを含まない。

10

20

30

40

50

イントラフレームは、デコーディングのためのエントリポイント、すなわち、デコーダが前フレームからの情報を使用せずにデコーディングを開始し得るポイントを提供するために、ビデオシーケンスに挿入され得る（例えば、定期的に、すなわち、特定の数のインターフレームの後ごとに）。

【0066】

イントラ予測ユニット190は、ブロック予測ユニットである。空間的または時間的予測を実行するために、符号化されたブロックは、逆量子化ユニット145および逆変換ユニット135によってさらに処理され得る。再構築器125によるブロックの再構築後、ループフィルタリングユニット160が、デコードされた画像の品質をさらに改善するために適用され得る。再構築器125は、再構築されたブロックを取得するために、デコードされた残差を予測子に加算する。次に、フィルタリングされたブロックは、参照フレームを形成し、参照フレームは、次にフレームバッファ170に記憶される。エンコーダ側のこのようなデコーディングループ（デコーダ）は、デコーダ側で再構築される参照ピクチャと同じ参照フレームを生成するという利点を提供する。したがって、エンコーダ側とデコーダ側は、対応する方法で動作する。ここでの「再構築」という用語は、デコードされた残差ブロックを予測ブロックに加算することによって、再構築されたブロックを取得することを指す。

【0067】

インター予測ユニット195は、インター符号化される現在のフレームまたはピクチャのブロックと、フレームバッファ170からの1つまたはいくつかの参照フレームまたはピクチャとを入力として受信する。動き推定および動き補償は、インター予測ユニット195によって実行される。動き推定は、例えばコスト関数に基づいて、動きベクトルおよび参照フレームを取得するために使用される。次に、動き補償は、参照フレームの参照ブロックの、現在のフレームへの転換の観点から、すなわち動きベクトルによって、現在のフレームの現在のブロックを記述する。インター予測ユニット195は、予測ブロックがコスト関数を最小化するように、1つまたはいくつかの参照フレーム内の候補ブロック（すなわち、候補予測子）のセットの中から現在のブロックの予測ブロック（すなわち、予測子）を選択する。言い換えれば、コスト関数が最小である候補ブロックが、現在のブロックの予測ブロックとして使用される。

【0068】

例えば、コスト関数は、現在のブロックと候補ブロックとの差の尺度、すなわち、候補ブロックに対する現在のブロックの残差の尺度であってもよい。例えば、コスト関数は、現在のブロックのすべてのピクセル（サンプル）と、候補参照ピクチャ内の候補ブロックのすべてのピクセルとの絶対差（SAD）の合計であってもよい。しかしながら、一般には、平均二乗誤差（MSE）または構造類似性メトリック（SSIM）などの類似性メトリックが用いられ得る。

【0069】

しかしながら、コスト関数はまた、そのようなインターブロックおよび／またはそのような符号化から生じる歪みを符号化するために必要なビットの数であってもよい。したがって、レート歪み最適化手順が、動きベクトルの選択、および／または一般に、ブロックに対してインター予測とイントラ予測のどちらを使用するか、およびその設定などのエンコーディングパラメータを決定するために使用されてもよい。

【0070】

イントラ予測ユニット190は、イントラ符号化される現在のフレームまたはピクチャのブロックと、現在のフレームの既に再構築された領域からの1つまたはいくつかの参照サンプルとを入力として受信する。次に、イントラ予測は、現在のフレームの参照サンプルの関数の観点から、現在のフレームの現在のブロックのピクセルを記述する。イントラ予測ユニット190は、現在のブロックの予測ブロックを出力し、前記予測ブロックは、好適には、符号化される現在のブロックとその予測ブロックとの差を最小化する、すなわち、それは残差ブロックを最小化する。残差ブロックの最小化は、例えばレート歪み最適化手順に基づいてもよい。特に、予測ブロックは、参照サンプルの方向補間として取得される

。方向は、レート歪み最適化によって、および／またはインター予測に関連して上で言及されたような類似性尺度を計算することによって決定され得る。

【0071】

次に、現在のブロックとその予測、すなわち残差ブロックとの差が、変換ユニット130によって変換される。変換係数は、量子化ユニット140によって量子化され、エントロピーエンコーディングユニット150によってエントロピー符号化される。このように生成されたエコードされたビデオビットストリームは、イントラ符号化ブロックおよびインター符号化ブロックならびに対応するシグナリング（モード指示、動きベクトルの指示、および／またはイントラ予測方向など）を含む。変換ユニット130は、離散フーリエ変換（DFT）または離散コサイン変換（DCT）などの線形変換を適用し得る。空間周波数領域へのこのような変換は、結果として得られる係数が通常、より低い周波数でより高い値を有するという利点を提供する。したがって、効果的な係数スキャン（ジグザグなど）および量子化の後、結果として得られる値のシーケンスは、通常、開始時にいくつかの大きな値を有し、ゼロの実行で終了する。これは、さらに効率的な符号化を可能にする。量子化ユニット140は、解像度の係数値を低下させることによって不可逆圧縮を実行する。次に、エントロピー符号化ユニット150は、バイナリコードワードを係数値に割り当てる。コードワードは、エコードされたビットストリームと呼ばれるビットストリームに書き込まれる。エントロピーコードはまた、上で示された分割フラグ構文による符号化を含み得るシグナリング情報（図1には示されていない）を符号化する。

【0072】

図2は、ビデオデコーダ200の例を示している。ビデオデコーダ200は、特に、参照ピクチャバッファ270と、ブロック予測ユニットであるイントラ予測ユニット290とを備える。参照ピクチャバッファ270は、エコードされたビデオビットストリームから再構築された少なくとも1つの参照フレームを記憶するように構成される。イントラ予測ユニット290は、デコードされるブロックの推定である予測ブロックを生成するように構成される。イントラ予測ユニット290は、参照ピクチャバッファ270から取得された参照サンプルに基づいてこの予測を生成するように構成される。

【0073】

デコーダ200は、ビデオエンコーダ100によって生成されたエコードされたビデオビットストリームをデコードするように構成され、好ましくは、デコーダ200とエンコーダ100の両方が、エコード／デコードされるそれぞれのブロックについて同一の予測を生成する。参照ピクチャバッファ270およびイントラ予測ユニット290の特徴は、図1の参照ピクチャバッファ170およびイントラ予測ユニット190の特徴と同様である。

【0074】

ビデオデコーダ200は、例えば逆量子化ユニット240、逆変換ユニット230、およびループフィルタリングユニット260のような、ビデオエンコーダ100にも存在するさらなるユニットを備え、これらは、それぞれ、ビデオエンコーダ100の逆量子化ユニット140、逆変換ユニット150、およびループフィルタリングユニット160に対応する。

【0075】

ビットストリーム解析、エントロピーデコーディング、および分割ユニット250は、量子化された残差変換係数およびシグナリング情報を取得するために、受信されたエコードされたビデオビットストリームを解析およびデコードするように構成される。量子化された残差変換係数は、残差ブロックを生成するために逆量子化ユニット240および逆変換ユニット230に供給される。残差ブロックは、再構築器225で予測ブロックに加算され、結果として得られた和は、デコードされたビデオブロックを取得するためにループフィルタリングユニット260に供給される。デコードされたビデオのフレームは、参照ピクチャバッファ270に記憶され、インター予測のための参照フレームとして機能し得る。ビットストリームから解析およびデコードされたシグナリング情報は、一般に、フレーム区分化に関連する制御情報を含み得る。画像をさらに正しく解析およびデコードするために、制御情報は、符号化ユニットへの画像の分割を復元して、次のデコードされたデータをそれ

それぞれの符号化ユニットに正しく割り当てるために使用される。

【0076】

一般に、図1および図2のイントラ予測ユニット190および290は、エンコードされる必要があるまたはデコードされる必要があるブロックの予測信号を生成するために、既にエンコードされた領域からの参照サンプルを使用し得る。

【0077】

ビットストリーム解析、エントロピーデコーディング、および分割ユニット250は、その入力として、エンコードされたビットストリームを受信する。ビットストリームが最初に解析され得る、すなわち、シグナリングパラメータおよび残差がビットストリームから抽出される。ビットストリームの構文およびセマンティクスは、エンコーダおよびデコーダが相互運用可能な方法で動作し得るように標準で定義され得る。

【0078】

HEVC規格では、スライス／ピクチャ境界に位置する符号化ツリーユニット（CTU）または符号化ユニット（CU）は、リーフノードの右下のサンプルがスライス／ピクチャ境界内に位置するまで、強制四分木分割（QT）を使用して分割される。強制QT区分は、ビットストリームでシグナリングされる必要はない。強制区分の目的は、エンコーダ／デコーダによって境界CTU／CUのエンコーディング／デコーディングを可能にすることである。言い換えれば、分割モードのさらなるシグナリングを必要とせずに、QT分割が使用されることがエンコーディング側およびデコーディング側で合意されている。

【0079】

QTBT構造に関する特許「国際公開第2016090568号」とVTM-1.0の両方で、境界CTU／CU強制区分プロセスは、HEVCから継承されている。これは、フレーム境界に位置するCTU／CU、特に、CTU／CUの一部がピクチャ／フレームの外部にあるように境界が通るCTU／CU（本開示では、このようなCTU／CUはそれぞれ「境界CTU」および「境界CU」とも呼ばれる）が、現在のCU全体がスライス／ピクチャ境界の内部に位置するまで、レート歪み（RD）最適化なしで四分木（QT）構造によって最初に強制的に区分化される。これらの強制区分は、ビットストリームでシグナリングされる必要はない。さらなる区分は、場合によりRD最適化に基づいて達成される。図6は、強制QTによるHD（1920×1080ピクセル）シーケンスの下部境界CTU（128×128）の1つの強制区分の例を示している。

【0080】

境界区分ではQT区分構造のみが使用されるため、VTMにおけるマルチタイプツリー（BT／TT／ABT）の制限は、SPSのMaxBTTDepthから示される。したがって、境界CTU（すなわち、図6に示されているように、境界の両側に部分を有するCTU）の区分化の場合、全体階層深度（TotalDepthLim）の制限、すなわち最大階層深度は、以下の式（1）に従う。

$$\text{TotalDepthLim} = \text{QTDepth} + \text{MaxBttDepth} \quad (1)$$

【0081】

式（1）で、MaxBttDepthは、SPSで指定される、許可された二分木および三分木最大深度である。QTDepthは、四分木区分化から生じる区分化ブロックの階層深度である。すなわち、四分木分割が適用される階層的区分化の各区分化ステップで、QTDepthの値は1だけ増加される。例えば、QTDepthは、表1の構文の例のパラメータcqtDepthに対応し得る。しかしながら、QTDepthの制限は各QT区分化ステップでインクリメントされるが、階層深度は、最終的に、所定の許可された最小CUサイズ（例えばSPSパラメータのminCUsSizeに対応する）によって制限される。

【0082】

ピクチャの境界は、BT、TT、またはABT（非対称二分木）を使用して処理され得る。強制的な方法と適応的な方法の両方が使用され得る。QT分割ではなくBT／TTなどのMTT（マルチタイプツリー）が境界区分で使用される場合、SPSからのBTT（二分木および三分木）の制限MaxBTTDepthは簡単に超えられてしまう。図7は、HDシーケンスの下部境

界のCTUで使用されるBTの例である。BTがCTUレベルから完全に境界の内部にあるリーフノード（符号化ユニット）まで使用され始める場合、4の深度がBT分割に使用され、これは、SPSからのMaxBTTDepth（VTM-1.0、VVCテストモデルバージョン1.0では、MaxBTTDepthは3に設定される）を超える。

【0083】

本開示の目的は、境界部分の区分の柔軟性を向上させ、ピクチャ境界処理の展望を提供することである。本開示のアプローチは、区分化に使用されるSPSからの従来技術のMaxBTTDepthに加えて、境界区分深度制限を定義することである。この目的のために、式(1)からのMaxBttDepthではなく、深度値が、境界区分化のために特に設定される（これは、例えばExtdMaxBTTDepthと呼ばれ得る）。ExtdMaxBTTDepthは、SPSで事前定義されて固定され、SPSビットストリームでシグナリングされ得るし、またはそれは計算され得る。したがって、マルチタイプツリー（MTT）分割（BT、TT、またはABTなど）がピクチャ境界処理（強制的または適応的）で使用される場合、最大MTT深度の制限は、デコーダとエンコーダの両方で拡張され得る。

【0084】

本開示の以下の態様および実施形態では、ExtdMaxBTTDepthを取得および計算する可能な方法が説明される。

【0085】

一般的な態様によれば、図8に示されているように、画像を符号化ユニットに分割するための装置800（上で説明されたエンコーダまたはデコーダのユニット110、250で実施され得る）が提供される。本装置は、画像を符号化ツリーユニットCTUに細分化する810ように構成された処理回路を含む。CTUは、水平方向および垂直方向に所定のサイズを有する非境界CTUと、水平または垂直の画像境界によって区切られた画像内の部分を有する境界CTUであって、その部分は、画像境界に垂直な方向の所定のサイズより小さいサイズを有する、境界CTUとを含む。回路は、非境界CTUおよび境界CTUをそれぞれの符号化ユニットに区分化するようにさらに構成される。その中で、非境界CTUの階層的区分化は、非境界マルチタイプ区分最大深度を有するマルチタイプ分割を含み、マルチタイプ分割は、分割方向が垂直方向または水平方向のいずれかである分割である。さらに、境界CTUの階層的区分化820は、境界マルチタイプ区分最大深度を有するマルチタイプ分割を含む。

【0086】

したがって、画像を符号化ユニットに分割するための装置800の回路は、区分化に含まれるMTT区分化ステップの第1の区分化深度制限、すなわち非境界マルチタイプ区分化最大深度を使用して非境界CTUを符号化ユニットに区分化するように構成される。回路は、区分化に含まれるMTT区分化の第2の区分化深度制限、すなわち境界マルチタイプ区分化最大深度を使用して、境界CTUを符号化ユニットに区分化する820ようにさらに構成される。言い換えれば、本開示によれば、装置800の回路は、動作中に、非境界CTUおよび境界CTUに対してそれぞれのMTT区分化最大深度（MTT深度制限）を使用する。

【0087】

フレーム分割810によって取得されたCTUは、境界CTUの階層的区分化820を含めて、さらに階層的に区分化され得る。この区分化は、例えば図3から図5に示され、上でこれらを参照して説明されたような方法で実行され得る。

【0088】

図8は、装置800の回路の内部構造を示している。この回路は、画像（またはフレーム）の、CTUへのそれぞれの分割（細分化）、非境界CTUを含むCTU、特に境界CTUの区分化のための機能ユニット810および820を有する任意の種類のハードウェアおよびソフトウェアであり得る。これらのユニットは、例えば単一のプロセッサ上で実施され得る。しかしながら、本発明は、このような適用に限定されず、これらのユニットは、別々のハードウェア部品によっても実施され得る。

【0089】

本開示において、「境界CTU」という用語は、画像境界によって、（画像境界内に）区

10

20

30

40

50

分化される画像の部分と、画像の内部に位置していない部分（すなわち、それは画像境界の向こうに位置する）とに分けられる符号化ツリーユニットを示すために使用される。符号化される画像のサイズが少なくとも一方向でCTUサイズの（整数）倍数でない場合に境界CTUは存在する。

【0090】

図9は、下部ピクチャ境界900および境界CTUを含む対応する境界部分910（影付き）の例を視覚化している。部分950は、垂直方向および水平方向にCTUの整数倍のサイズを有する画像の残りの部分を示している。さらに、CTUの垂直サイズは970Vとして示され、CTUの水平サイズは970Hとして示されている。図9において理解され得るように、この例の境界部分は、水平方向ではCTUサイズ970Hの整数倍である。しかしながら、垂直方向では、境界部分910は、垂直CTUサイズ970Vによる垂直ピクチャサイズの分割後の残りのサイズを有する。部分920は、仮想にすぎず、境界部分の高さとCTUサイズとの差を示している。本実施態様では、CTUは正方形であり、このため、サイズ970Hと970Vは同じであることに留意されたい。しかしながら、本開示はこれに限定されず、CTUの垂直サイズと水平サイズは異なってもよい。

【0091】

境界部分をエンコード（およびこれに対応してデコード）するために、図9の境界部分910は、不完全なCTUに、すなわち、CTU970Hの水平サイズおよびCTUサイズ970Vより小さい垂直サイズを有するCTU部分に分割される。これらの不完全なCTUは、本出願の「境界CTU」に対応し、「不完全なCTU」は、画像内のCTUの部分に対応する。図9には水平境界部分が示されているが、水平境界部分に加えて、またはその代わりに、垂直境界部分が存在する場合もある。特に、垂直方向および水平方向のどちらでも画像サイズ（すなわち、画像の幅および高さ）がCTUサイズの倍数でない場合、水平境界および垂直境界のCTUである不完全なCTUがさらに存在し、画像内のCTU部分の垂直サイズおよび水平サイズは、完全なCTUのサイズより小さい。

【0092】

「境界CTU」とは対照的に、「非境界CTU」という用語は、フレームまたはピクチャに完全に適合するCTU、すなわち、図9に示されている例では部分950の内部のCTUを示すために使用される。すなわち、非境界CTUは、垂直方向および水平方向においてその完全なサイズでエンコード／デコードされる画像の内部に位置する。しかしながら、非境界CTUは、少なくとも1つの画像境界に隣接する場合もあれば、またはそれらは、非境界CTUと画像境界との間のさらなるCTUである場合もある。

【0093】

非境界CTUおよび境界CTUは、符号化または指定される方法が互いに異なる異なる種類のCTUを構成しないことにさらに留意されたい。非境界CTUと境界CTUとの差は、それらが境界（すなわち、それらを通る境界）に位置するのか、それとも画像を区切る境界の内部に位置するのかわかる。

【0094】

さらに、CTUが境界CTUであるかどうかは、例えば、CTUの位置（特に、CTU内の適切なピクセル位置）と境界の位置（またはサンプルの垂直／水平画像サイズ）とを比較することによって判定される。符号において、CTUは、事前定義された固定サイズ、上で説明されたように、例えばHEVCの場合のように128×128または64×64を有する。ピクチャは、重複なしでCTUに分割される。適切なピクセル位置として、右下隅が選択され得るが、これは、画像が左から右へ、また上から下へ処理される場合に、CTUが垂直ピクチャ境界および水平ピクチャ境界のそれぞれに対して境界CTUであるとそれが判定させるためである（一般に、どのピクセル位置が最適かは処理方向に依存する）。エンコーダ／デコーダは、CTUの右下隅のピクセルをチェックし、それと画像の垂直サイズおよび水平サイズとを比較する。右下のピクセルがピクチャ境界の内部に位置する場合、CTUは非境界CTUであり、そうでなければ、それは境界CTUである。区分化ブロックが境界上に位置するかどうかを判定する方法は、CTUに適用されるだけでなく、CTUまたは区分化階層内の

何らかのブロックを分割することから生じるCUおよび任意の区分ブロックにも使用され得る。

【0095】

さらに、CTUが境界CTUであるかどうかを判定する他の方法が使用されてもよい。例えば、上で言及されたように、CTUサイズによる画像サイズ（幅／高さ）の分割は、CTUの数と、垂直方向と水平方向に境界CTUが存在するかどうかとを判定し得る。CTUにはインデックスが付けられ得、境界CTUが存在する場合、最後のk個のCTUが、下部境界CTUであると判定され得るか、または各k番目のCTUが、右側境界CTUに対応し得る（kは、一行あたりのCTUの数（境界CTUを含む）、すなわち、画像幅とCTUサイズの比の上限である）。

10

【0096】

階層的区分化の階層深度は、図3に関して上で説明されている。したがって、区分深度は、特定の区分化レベルを取得するために実行される区分化ステップの数に対応し、レベル0（または層0）に対応する区分化深度0を有するCTUから始まり、符号化ユニットの深度まで達する。

【0097】

非境界CTUの場合、符号化ユニットの階層深度は、それがSPSの設定などの設定に従う最大階層深度を超え得ないように制限される。例えば、設定では、四分木分割とマルチタイプ分割のそれぞれの区分化深度に対して異なる制限が設定され得る。例えば、CTUの区分化は、深度制限がQT区分化ステップとMTT分割に対して別々に設定され得るため、異なるレベルまたは同じレベルの異なる部分ブロックにQT分割とMTT分割（BT、TT、およびABT分割など）の両方を含む。例えば、非境界MTT最大深度は、現在のVTMのパラメータMaxBTTDepthに対応してもよいし、またはそれは、例えばMaxMTTDepthと、再定義され名付けられ、BT、TTなどのすべてのMTT分割タイプに適用されてもよい。したがって、深度制限、すなわちCTU分割の最大全体深度は、QT分割およびMTT（例えば、BTT）のそれぞれの深度制限の合計であり得る。

20

【0098】

一方、境界CTUに関して、本開示は、以下の式（2）によって与えられる深度制限を提供する。

$$\text{TotalBPDepthLim} = \text{QTDepth} + \text{ExtdMaxBTTDepth} \quad (2)$$

30

【0099】

式（2）から理解され得るように、本開示では境界区分深度とも呼ばれる、境界CTUの深度制限は、一般に、QT分割の深度およびMTT分割（これは二分分割および三分分割に限定されず、ExtdMaxBttDepthという名称は単なる例と見なされるべきである）の深度から構成される。QT深度は、強制QT分割の場合に式（1）で使用されるのと同じQT深度であり得る。しかしながら、式（2）は、代替的には例えば「ExtdMaxMTTDepth」と呼ばれ得る、境界マルチタイプ区分最大深度ExtdMaxBTTDepth（拡張された二分木／三分木最大深度）によって式（1）とは異なる。

【0100】

装置800に対応して、本開示のさらなる態様として、画像を符号化ユニットに区分化するための方法が提供される。本方法は、画像を符号化ツリーユニット（CTU）に細分化するステップを含む。その中で、CTUは、水平方向および垂直方向に所定のサイズを有する非境界CTUと、水平または垂直の画像境界によって区切られた画像内の部分を有する境界CTUであって、その部分は、画像境界に垂直な方向の所定のサイズより小さいサイズを有する、境界CTUとを含む。本方法は、非境界CTUおよび境界CTUをそれぞれの符号化ユニットに階層的に区分化するステップをさらに含む。その中で、非境界CTUの階層的区分化は、非境界マルチタイプ区分最大深度を有するマルチタイプ分割を含み、マルチタイプ分割は、分割方向が垂直方向または水平方向のいずれかである分割である。境界CTUの階層的区分化は、境界マルチタイプ区分最大深度を有するマルチタイプ分割を含む。

40

【0101】

50

以下では、画像を符号化ユニットに分割するための装置800および対応する方法の両方をさらに明示する、本開示のいくつかの例示的な実施形態が説明される。

【0102】

実施形態1

上で説明されたように、HEVCまたはVTM-1.0は境界CTUの区分化に強制QTを使用するため、境界区分化のために別個のBTT深度制限を定義する必要はない。HEVCまたはVTM-1.0では、式(1)からのパラメータMaxBttDepthは、境界の場合と非境界の場合の両方に適用される。

【0103】

しかしながら、CE1は、境界区分としてBTまたはTTを含み始めるため、SubCE2に関連して上で説明された理由により、式(1)による境界のMaxBTTDepthの制限は適切でないことがない場合がある。実施形態1では、MTT区分構造が境界区分に含まれる場合に境界区分にのみ使用される新しいMaxBTTDepthが定義される。

【0104】

本開示は、ハイブリッドビデオ符号化における境界区分MTT深度の新しい制限の定義を提供する。言及されたように、この境界マルチタイプ区分最大深度は、計算も事前定義もされ得る。本実施形態1は、境界マルチタイプ区分最大深度を計算するいくつかの例示的な方法を提供する。

【0105】

実施形態1では、境界マルチタイプ区分最大深度は、少なくとも適応境界マルチタイプ区分深度と事前定義されたマルチタイプ区分深度との合計である。適応境界マルチタイプ区分深度は、分割方向が画像境界の方向であるマルチタイプ分割の深度である。

【0106】

したがって、境界ブロック区分の柔軟性を維持するために、境界CTUに位置するブロックの全体深度制限の公平な処理を実行することが提案され、公平性は、式(1)によって定義されるようなJEMピクチャ境界処理におけるQT分割と比較して、まさに境界CTUのMTT分割である。

【0107】

この目的のために、式(2)で使用される境界MTT区分最大深度は以下のように再定義される。

$$\text{ExtdMaxBTTDepth} = \text{BTTBPPDepth} + \text{MaxBTTDepth} \quad (3)$$

【0108】

その中で、ExtdMaxBTTDepthとして示される境界MTT区分最大深度は、式(1)または(2)から導出され得る。したがって、境界CTUの全体区分化深度制限は以下のように再定義される。

$$\text{TotalBPPDepthLim} = \text{QTDepth} + (\text{BTTBPPDepth} + \text{MaxBTTDepth}) \quad (4)$$

【0109】

式(3)および(4)において、本実施形態1の適応境界マルチタイプ区分深度であるBTTBPPDepthは、各層で決定され、これにより、ExtdMaxBTTDepthも、区分化層、すなわち各区分化ステップの符号化ブロックの層と、最終的に符号化およびデコードされる結果のCUの層とに応じて変化する。例えば、適応境界マルチタイプ区分深度は、BT/TT/ABT境界区分化などの異なるMTT分割モードで同じになるように選択され得る。適応境界マルチタイプ区分深度は、各区分化層において、この層でBT、TT、ABT、または何らかの他のMTT区分化タイプのどれが実行されるかに関係なく、再決定される(すなわち、増加される)。QT分割の場合と同様に、絶対深度制限は、所定の最小CUサイズ、例えば、上で言及されたSPSパラメータminCUSizeから生じ得る。BT区分化ステップの場合、深度は、ステップから生じる各区分ブロックで1だけ増加される。TT区分化の場合、VTM-1.0に従って、深度は、符号化ユニットの深度とサイズの関係を確認するために、またBT分割とTT分割との間の互換性を確保するために、結果の1/2サイズのブロックでは1だけ増加され、1/4サイズのブロックでは2だけ増加される。

【0110】

上で言及されたように、適応マルチタイプ区分深度は、分割方向が画像境界の方向である深度である。これは、対応する層のブロックを、後のより深い層の2つ、3つ、またはそれ以上のブロックに分割する分割線が分割される境界CTUを通る当該の画像境界と同じ方向を有する、すなわち、平行であることを意味する。しかしながら、当該のピクチャ境界に垂直な分割方向での分割／区分化ステップは、適応マルチタイプ区分深度の値に寄与しない。特に、ピクチャ境界が水平境界である場合、水平分割方向のステップのみが、適応マルチタイプ区分深度によってカウントされる。そうではなく、ピクチャ境界が垂直境界である場合、分割方向が垂直であるステップのみがカウントされる。これは、図10および図11に示されている区分化の例で理解される。

10

【0111】

しかしながら、本開示は、水平境界／例えば、下部境界）と垂直境界（例えば、右側境界）の両方に位置する境界CTU、例えば右下CTUの区分化階層内の境界CUおよび境界区分ブロックにも適用可能である。この場合、両方の分割方向のステップMTTステップは、式（3）のBTTBPDDepthの項によって、あるいは別個のそれぞれの変数によってカウントされる。

【0112】

一方、式（3）および（4）から理解され得るように、境界マルチタイプ区分最大深度における非適応項（すなわち、事前定義されたマルチタイプ区分深度）は、例えば、非境界マルチタイプ区分最大深度に等しくてもよい。しかしながら、本開示はこれに限定されない。例えば、追加の事前定義されたマルチタイプ境界区分深度パラメータが定義されるべき場合、（例えば、1つ以上のビットを節約するために）このパラメータをより小さな値に設定することが有益であり得る。

20

【0113】

図10は、二分木区分化を使用する境界区分化の第1の例を、特に、BTを使用する下部境界区分化の例を示している。図の上半分の左側に示されている開始ポイントは、境界CTUと、境界CTUを通る下部ピクチャ境界（太い実線）とを示している。さらに、CTUの右上隅には、画像分割によって取得されるマークされたブロックオブジェクトがある。図の上半分の右側は、境界CTUが区分化される最終的な区分化（すなわち、区分化パターン）を示している。ただし、この第1の例および図10に示されている例では、MTT（特にBTT）の区分化ステップのみが考慮されている。式（4）に従って、QT分割は、MTT区分化ステップに先行するさらなるステップで実行され得る。したがって、開始ポイントは、必ずしもCTUである必要はなく、例えば、QT分割から生じる0より大きい深度の正方形の符号化ブロック（垂直サイズおよび水平のサイズが等しい）であってもよい。

30

【0114】

図10の下半分には、開始ポイントから最終的な区分化まで境界CTUを区分化する区分化ステップ1から3が示されている。その中で、太い実線はピクチャ境界であり、BTT境界およびCTU区分は、実線（境界の内部）または破線（境界の外部）として示されており、点線（ステップ3の）は、BTTの「通常の」区分である（すなわち、垂直である区分は、分割方向が画像境界の方向である境界区分ではない）。式（1）に従ってMaxBTTDepth（VTMでは制限は3）が使用された場合、3つのMTT区分化ステップがあるため、CTUの右上のオブジェクトブロックはさらに区分化され得ず、したがって、CurrentBTTDepth（3） $\geq$ MaxBTTDepth（3）（CurrentBTTDepthは、BTT区分化が実行されるステップの数に対応する、現在のステップでのBTT区分化の深度である）である。

40

【0115】

実施形態1によれば、ステップ1では、（BTT）境界区分化（すなわち、分割方向が境界の方向である区分化）の1つの層が存在する。したがって、BTTBPDDepth=1である。したがって、式（3）から、ExtMaxBTTdepth=BTTBPDDepth+MaxBTTDepth=1+3=4が得られる（SPSではMaxBTTDepthは3に設定される）。ステップ1の最大BTT深度層（境界マルチタイプ区分最大深度）は4であり、ブロックは既に1回区分化されているた

50

め、区分化されたブロックは、境界マルチタイプ区分最大深度に従ってステップ1で設定されたExtMaxBTTdepthに従って、さらに3回MTT区分化され得る。

【0116】

ステップ2では、これまでに実行されたBP（境界区分化）によって取得された境界区分化の2つの層があり、BTTBPDepth=2である。したがって、式(3)から、 $\text{ExtMaxBTTdepth} = \text{BTTBPDepth} + \text{MaxBTTDepth} = 2 + 3 = 5$ が得られる（SPSではMaxBTTDepthは3に設定される）。ステップ2の新しい最大BTT深度層（境界マルチタイプ区分最大深度）は5（開始ポイント（a）から計算された）であり、ブロックは既に2回区分化されているため、区分化されたブロックは、境界マルチタイプ区分最大深度に従って、さらに3回区分化され得る。境界区分化（すなわち、分割方向は水平であり、画像境界方向と同じ方向である）、BTTBPDepth、したがってExtMaxBttDepthがステップ1からステップ2にかけて1だけ増加されたことが理解され得る。

10

【0117】

ステップ3では、さらなるBT区分化が実行されている。ただし、この最後の区分化は、画像境界の方向ではない分割方向を有し、特に、ステップ3の分割方向は、画像境界に垂直である。このため、ステップ3はBPPBPDepthの値に寄与しない。したがって、BTTBPの2つの層がまだあり、BTTBPDepth=2である。以下のように、 $\text{ExtMaxBTTdepth} = \text{BTTBPDepth} + \text{MaxBTTDepth} = 2 + 3 = 5$ （SPSではMaxBTTDepthは3に設定される）である。説明されたように、適応マルチタイプ区分深度では、分割方向が画像境界方向であるマルチタイプ分割の深度（ステップ／層の数）のみが寄与する。しかしながら、境界マルチタイプ区分最大深度は、MTT境界区分化の制限である。したがって、ステップ3の最大深度層は5（開始ポイントから計算された）のままであり、ブロックは既に3回区分化されているため（ここでは、BPだけでなくすべての区分が考慮されている）、区分化されたブロックはさらに2回BTT区分化され得る。

20

【0118】

図11には、MTT区分最大深度計算の第2の例が示されている。この図では、図10と同様に、太い実線はピクチャ境界であり、実線（内部）または破線（外部）は、CTU境界またはBTT境界区分である。境界CTUの内部に位置するマークされた対象（すなわち、MTT分割によって取得されるターゲット符号化ブロック）がある。図の右上部分の最終的な区分パターンを取得するには、図のステップ1から4に示されているように、4つの区分化のステップが実行される必要がある。

30

【0119】

MaxBTTDepthが式1に従って使用され、VTMの場合のように3に制限される場合、 $\text{CurrentBTTDepth} (4) \geq \text{MaxBTTDepth} (3)$ であるため、マークされたターゲットブロック（MTT深度4の）は取得され得ず、さらに区分化され得ない。

【0120】

図11のステップ1では、BTTBPの1つの層があり、BTTBPDepth=1である。したがって、 $\text{ExtMaxBTTdepth} = \text{BTTBPDepth} + \text{MaxBTTDepth} = 1 + 3 = 4$ （SPSではMaxBTTDepthは3に設定される）である。ステップ1の最大BTT深度層は4であり、ブロックは既に1回区分化されているため、区分化されたブロックはさらに3回BTT区分化され得る。

40

【0121】

図11のステップ2では、BTTBPの2つの層があり、BTTBPDepth=2である。したがって、 $\text{ExtMaxBTTdepth} = \text{BTTBPDepth} + \text{MaxBTTDepth} = 2 + 3 = 5$ である。ステップ2の最大BTT深度層は5であり、ブロックは既に1回区分化されているため、区分化されたブロックはさらに3回BTT区分化され得る。

【0122】

図11のステップ3では、BTTBPの3つの層があり、BTTBPDepth=3である。したがって、 $\text{ExtMaxBTTdepth} = \text{BTTBPDepth} + \text{MaxBTTDepth} = 3 + 3 = 6$ である。ステップ2の最大BTT深度層は6であり、ブロックは既に1回区分化されているため、区分化されたブロックはさらに3回BTT区分化され得る。

50

【0123】

さらに、ステップ4では、BTT区分化の4つの層がある。 $\text{ExtdMaxBTTdepth} = \text{BTTBPD} + \text{MaxBTTDepth} = 4 + 3 = 7$ である。ステップ4の最大深度層は7（開始ポイント（a）から計算された）であり、ブロックは既に4回区分化されているため、区分化されたブロックはさらに最大で3回区分化され得る。

【0124】

境界マルチタイプ区分最大深度では、分割方向が境界の方向であるBPだけでなく、すべての区分が考慮され制限される。しかしながら、図11に示されている例のステップのすべてにおいて、分割方向が画像境界の方向であるマルチタイプ分割が実行されている。したがって、これらのステップのそれぞれでは、適応境界マルチタイプ区分深度（この例ではBTTBPD）の値が1だけ増加される。

10

【0125】

境界マルチタイプ区分化の深度制限を、分割方向が境界と同じ方向である区分化ステップの数に適應させることによって、本開示は、より实际的に画像境界の近傍のCTUの残りの部分に近似する符号化ユニットの区分化パターンを容易にする。特に、QT分割のみを使用するのと対照的に、境界CTU／部分の符号化ユニットの数が低減され得る。

【0126】

実施形態2

上で説明された実施形態1によれば、境界マルチタイプ区分最大深度（例えば、 ExtdMaxBTTDepth として与えられる）は、少なくとも適応境界マルチタイプ区分深度と事前定義されたマルチタイプ区分深度との合計である。ここで、本実施形態2によれば、この合計は、境界CTUの境界区分ブロックの画像境界の方向と画像境界に垂直な方向とのサイズの比の関数をさらに含む。その中で、境界区分ブロックは、適応境界マルチタイプ区分深度のブロックである。

20

【0127】

本開示では、「区分ブロック」という用語は、より低いレベルのブロック／ユニットを分割することから生じるCTUまたはブロックを指す。したがって、区分ブロックは、エンコード／デコードされる階層区分化の最終結果である階層的区分化または符号化ユニットの最上部のCTUに限定されず、図10のステップ1および2または図11のステップ1から3などの中間分割ステップで取得されるブロックも含む。さらに、「境界区分ブロック」は、境界CTUが境界に位置するのと同様に、画像境界（ブロックを通る画像境界）に位置する区分ブロックである。

30

【0128】

当該の境界が水平画像境界である場合、上で言及されたサイズの比は、分割された境界区分ブロックの水平サイズ（幅）と垂直サイズ（高さ）の比である。一方、画像境界が垂直である場合、この比は、境界区分ブロックの幅で高さを割ったものである。

【0129】

例えば、比の関数は、2進対数 $\log_2$ 比であってもよい。実施形態1では、全体深度制限は、境界ブロックのBTT制限を定義する基準と見なされる。実施形態2は、式1を使用するMTTBPD深度制限の計算と比較して、特に、公平性基準（すなわち、QT分割とMTT分割との間の公平性）として符号化ユニットの到達可能なサイズを考慮する。したがって、境界マルチタイプ区分最大深度は、以下の式（5）に従って拡張される。

40

$$\text{ExtdMaxBTTDepth} = \log_2 \text{Ratio} + \text{BTTBPD} + \text{MaxBTTDepth} \quad (5)$$

ただし、Ratioは幅と高さの比（境界が水平である場合は幅／高さ、または境界の高さ／幅は垂直である）を示す。

【0130】

実施形態2による境界区分化の例が図12に示されている。図のパート（a）は、強制QTを使用したVTM-1.0の境界区分化を示しており、一方、図1のパート（b）は、BTを使用した境界区分である。図10および図11のように、実線（境界の内部）。破線は、CTUおよび境界区分化（分割方向は境界方向である）を示し、点線は、分割方向が境界方向と

50

同じでない区分化を示している。図12のパート (a) では、SPSからのMaxBTTDepthがVTM-1.0の構成で3に設定されているため、CTUの左上のマークされたブロックは、可能性としてBTの3つの追加レベルでさらに分割され得、これまでBT分割は実行されていない。パート (b) では、CTUの左上のマークされたブロックをさらに3つのレベルのBT (または他のMTTタイプ) で区分化することも可能あり、したがって、マークされたブロックのExtdMaxBTTDepthは、以下のような式 (5) から取得される。

$$\text{ExtdMaxBTTDepth} = \log_2 (\text{boundary partition block ratio (4)}) + \text{BTTBPdepth (2)} + \text{MaxBTTDepth (3)} = 7$$

【0131】

図12のパート (b) に示されている現在の実施形態2の場合、境界マルチタイプ区分最大深度を示すExtendedMaxBTTDepthに従って、マークされたブロックは、可能性としてBTのさらに3つのレベルに分割され得る (ExtdMaxBTTDepth (7) - CurrentBTTDepth (4)) = 3。

【0132】

ここでは、マークされたブロックは、分割方向が境界方向である2つのBT区分ステップと、それに続く、分割方向が境界区分によって垂直である2つのBTステップによって得られたと想定されており、結果の現在のBT深度は4に等しい。しかしながら、本開示は、BT分割であるMTT分割に限定されない。例えば、区分ブロックが同じ分割方向に3回以上分割される単一のステップで4つの区分ブロックが取得される分割モードがさらに使用され得る (4つの区分化ブロックが取得されるこのような分割モードは、場合によりSplitInto4と呼ばれ、これは、同様に1つのステップで4つの区分ブロックが取得されるがQT分割とは異なる)。図12の例で、最後の2つのBT分割ステップが1つのSplitInto4分割ステップに置き換えられた場合、結果の現在のMTT深度は、4ではなく3である。

【0133】

さらに、MTTでは、カウントされた階層深度は、結果として得られる最小の区分の結果のブロックサイズに関して決定され得る。例えば、VTM-1.0の現在のTTの実施態様では、ブロックは、1/4、1/2、および1/4のサブブロック/区分ブロックに分割される。第1および第3の区分ブロックでは、深度がBTとTTの両方で等しく有効であることを確認するために、VTM-1.0では深度は2回カウントされる (また、結果の区分ブロックの小さい方でABTステップが2回カウントされ得る)。したがって、BT分割との互換性を確保するために、単一のSplitInto4ステップも2つのステップとしてカウントされ得る。このようにして、区分ブロック/符号化ユニットの結果のブロックサイズは深度から理解され得る。しかしながら、他のいくつかの可能な方法では、例えばいくつかの変数から、それぞれの区分化タイプが知られることが保証される場合、各ステップは1回カウントされ得る。

【0134】

式 (5) において、特にBT区分化が適用される場合 (図12のパート (b) に示されているように)、右側の境界区分ブロック比の2進対数 (第1の項) は、MTT BP深度 (第2の項) と同じであり得ることに留意されたい。

【0135】

本開示は、2進対数である境界区分ブロックの関数に限定されないことにさらに留意されたい。例えば、定量自体 (恒等関数) が使用されてもよいし、または例えば、比に定数または何らかの適応パラメータが掛けられてもよい。

【0136】

実施形態3

上で説明された実施形態1および2では、境界マルチタイプ区分最大深度がどのように計算され得るかが説明された。一方、本実施形態3では、境界マルチタイプ区分最大深度は事前定義される (すなわち、それは固定される)。所定の値は、式 (4) の項BTTBPDepthとして提供され得るか、またはそれは、境界CTUについて式 (1) のような式のMaxBttDepthに取って代わり得る。

【0137】

例えば、このような固定された境界マルチタイプ区分最大深度は、ビットストリームのシグナリングから取得され得る。例えば、それは、HEVCにおけるPPS、SPS、VPSなどの1つ以上のビデオフレームに共通の制御パラメータのセット内でシグナリングされ得る。それはまた、ビデオシーケンス全体ごとに1回シグナリングされ得る。シグナリングは、ビットストリームへの差分符号化、予測符号化、エントロピー符号化、または任意の他の埋め込みなどのパラメータのさらなる符号化を含み得る。

【0138】

しかしながら、説明されたように、上記の実施形態1から3を含む本開示は、固定され、ビットストリームでシグナリングされる境界マルチタイプ区分最大深度に限定されない。それはまた、例えば、ビットストリームで伝達される1つ以上のパラメータ（SPSのMaxBTTDepthなど）および／または上で言及されたBTTBPDepthまたはQTDepthなどの内部変数に基づいて、導出され得る。しかしながら、導出はまた、標準のまたはシグナルされた事前定義された関係に基づいてもよい。

【0139】

いくつかの例（式1から5を含む）の説明はBTT区分化に言及しており、変数はそれに応じて名付けられている（例えば、BTTBPDepth、MaxBTTDepth）が、本開示は、ABT（非対称二分区分化）区分化（区分化ブロックは、1つの区分化ステップで、 $1/4$ および $3/4$ などの異なるサイズを有する2つ以上の区分ブロックに区分化される）または図12のパート（b）に関連して言及されたSplitInto4Modeなどの他のタイプのMTT区分化にも適用可能である。したがって、変数は、例えばBTTBPDepth、MaxBTTDepthなど異なつて名付けられ得る。したがって、本発明は、使用される特定のMTT分割モードに関係なく、QT分割に加えて代替のマルチタイプ分割モードを使用するピクチャ境界処理のために柔軟性を提供することを容易にする。

【0140】

しかしながら、いくつかの実施形態では、MTT分割に加えて、境界CTUの階層分割（および非境界CTUの階層分割）は、QT分割をさらに含み得る。これは、境界マルチタイプ区分最大深度（括弧内の項）とQT区分深度（内部変数QTDepth）の合計として全体境界区分最大深度を定義する式（4）から理解され得る。例えば、VTM-1.0の場合と同様に、クォータナリ分割は、MTT分割の前に実行され得る。すなわち、QT分割ステップが実行され、それに続いてMTT分割ステップが実行される。しかしながら、本開示はこれに限定されず、それは、MTT分割後にQT分割が実行される構成にも適用可能である。したがって、言及されたように、図10および図11の開始ポイントは、CTUである必要はなく、1つ以上の前の区分化ステップから取得された区分ブロックであってもよい。

【0141】

さらに、本開示によれば、境界マルチタイプ区分最大深度は、非境界マルチタイプ区分最大深度以上であり得る。例えば、境界CTUで、分割方向が境界の方向であるマルチタイプ分割が実行されない場合、境界マルチタイプ区分最大深度は、非境界マルチタイプ区分最大深度に等しくなり得る。この場合、内部変数BTTBPDepth（またはMTTBPDepth）がインクリメントされる区分化ステップはない。それでも、画像を符号化ユニットに分割するための装置は、説明されたように、それぞれ異なる方法で、非境界マルチタイプ区分深度および境界マルチタイプ区分最大深度の決定を実行する。

【0142】

しかしながら、境界マルチタイプ区分最大深度が事前に決定される実施形態3では、境界CTUにおけるMTT区分化のより高い柔軟性を可能にするために、その事前に決定される値は、非境界マルチタイプ区分最大深度より大きくなければならない。

【0143】

本開示のBT/TT（またはMTT一般の）最大深度拡張のフローチャートが図13に示されている。条件の「境界にCTUはあるか」は、現在のCTU/CUが境界に位置するCTUの内部に位置するかどうかという条件を示す。この条件が偽（N）である場合、可能なBT/TT（

10

20

30

40

50

または他のMTT) 最大深度を制限するために、SPSからの通常のMaxBTTDepthが非境界マルチタイプ区分深度として使用される。そうでなければ、MaxBttdepthは、本開示の実施形態のいずれかに従って拡張および決定される。

【0144】

言及されたように、画像を分割するための装置800は、図1および図2に示されているようなビデオエンコーディング装置100またはデコーディング装置200に組み込まれ得る。したがって、本発明は、ビデオシーケンスの画像をエンコードするための装置100、すなわちエンコーディング装置をさらに提供する。エンコーディング装置100は、本開示で説明されている実施形態のいずれかに従って画像を分割するための装置800と、符号化ユニットをエンコードする画像符号化ユニットと、エンコードされた符号化ユニット、および符号化ツリーユニットがどのように区分化されたかを示す区分化情報を含むビットストリームを生成するように構成されたビットストリーム形成ユニットとを備える。

10

【0145】

特に、区分情報は、各CTUについて、または中間ステップの区分化ブロックについて、QTおよびいくつかのMTT分割モードの中からどの区分化タイプモードが適用されているかを示す情報を含み得る。例えば、区分ブロックまたはCTUの各分割について、分割モードを示すパラメータが、ビットストリームに含まれてもよい。代替的に、特定の分割モードが強制的に使用されてもよい。

【0146】

境界CTUと非境界CTUの区分化モードが異なる場合があることに留意されたい。例えば、上で例示されたように、区分化モードは、境界CTUと非境界CTUの両方についてCTUベースで（および／または区分ブロックベースで）シグナリングされ得る。代替的に、特定の境界（垂直または水平）のすべてのCTUの区分化モードは、同じであり、1つ以上のビデオピクチャに関連するシグナリング内に設定されてもよい。

20

【0147】

しかしながら、境界CTUの区分化モードは、非境界CTUの区分化モードとは異なる方法で決定されてもよい。例えば、境界CTU（および区分ブロック）に関して、事前に決定された区分化モードが強制されてもよい、すなわち、規格は、固定区分化モード、または境界CTUの区分化モードを、例えば、隣接するCTUの境界および／またはサイズおよび／または区分化モード内のその位置に基づいて決定するためのアルゴリズムを定義してもよい。

30

【0148】

これに対応して、本開示は、ビデオシーケンスの画像をエンコードするための方法をさらに提供する。本方法は、本開示で説明されている実施形態のいずれかに従って画像を分割する方法のステップと、符号化ユニットをエンコードする画像符号化ステップと、エンコードされた符号化ユニット、および符号化ツリーユニットがどのように区分化されたかを示す区分化情報を含むビットストリームを生成するビットストリーム形成ステップとを含む。

【0149】

また、ビデオシーケンスの画像をデコードするための装置200および方法も提供される。デコーディング装置200に組み込まれる場合、装置800は、（デコードされる）画像の、符号化ユニットへの分割を決定するために使用される。デコーディング装置200は、エンコードされた符号化ユニットを含むビットストリームを解析するためのビットストリームパーサと、実施形態のいずれかによる画像の分割を決定するための装置800と、決定された画像の分割に基づいて、エンコードされた符号化ユニットをデコードするための画像デコーディングユニットとを含む。

40

【0150】

これに対応して、画像をデコードするための方法は、エンコードされた符号化ユニットを含むビットストリームを解析するステップと、本開示の実施形態のいずれかによる画像の分割を決定するステップと、画像の決定された分割に基づいて、エンコードされた符号化ユニットをデコードする画像デコーディングステップとを含む。

50

【0151】

デコーダ側では、最大BT/TT深度が、境界CTU/CUで拡張される、すなわち、境界マルチタイプ区分最大深度が、デコードされる境界CTUに使用される。拡張は、SPSから単純に解析されてもよいし、または特定の条件に基づいて導出されてもよい。実施形態1に基づいて、最大BTT深度は、BTT境界区分深度で拡張され得る。また、実施形態2に基づいて、最大BTT深度は、境界区分ブロックの比およびBTT境界区分深度で拡張されてもよい。

【0152】

エンコーダ/デコーダおよびそれぞれのエンコーディング/デコーディング方法によるビデオ画像のエンコーディングおよびデコーディングにおいて、分割（または分割の決定）が実施形態3に従って実行される場合、境界マルチタイプ区分最大深度は事前定義される。この場合、エンコード/デコードされたビットストリームは、境界マルチタイプ区分化最大深度を含むエンコードシーケンスパラメータセットをさらに含み得る。デコーダ側では、画像の分割を決定するための装置800は、次に、シーケンスパラメータセットから第2のマルチタイプ区分化最大深度を取得するようにさらに構成され得る。

【0153】

図14は、符号化システム300、例えばピクチャ符号化システム300の一実施形態を示す概念的または概略的なブロック図であり、符号化システム300は、エンコードされたデータ330、例えばエンコードされたピクチャ330を、例えば、エンコードされたデータ330をデコードするための宛先デバイス320に提供するように構成されたソースデバイス310を備える。

【0154】

ソースデバイス310は、エンコーダ100またはエンコーディングユニット100を備え、追加的に、すなわち任意選択で、ピクチャソース312、前処理ユニット314、例えばピクチャ前処理ユニット314、および通信インターフェースまたは通信ユニット318を備え得る。

【0155】

ピクチャソース312は、例えば実世界のピクチャを取り込むための任意の種類のピクチャ取り込みデバイス、ならびに/または任意の種類のピクチャ生成デバイス、例えば、コンピュータアニメーションピクチャを生成するためのコンピュータグラフィックプロセッサ、または実世界のピクチャ、コンピュータアニメーションピクチャ（例えば、画面コンテンツ、仮想現実（VR）ピクチャ）、および/もしくはこれらの組み合わせ（例えば、拡張現実（AR）ピクチャ）を取得および/もしくは提供するための任意の種類のデバイスを含み得るか、またはこれらであり得る。以下では、これらの種類のすべてのピクチャおよび任意の他の種類のピクチャは、特に明記しない限り、「ピクチャ」または「画像」と呼ばれ、「ビデオピクチャ」および「静止ピクチャ」を包含する「ピクチャ」という用語に関するこれまでの説明も、明示的に別段の指定がない限り、依然として当てはまる。

【0156】

（デジタル）ピクチャは、強度値を有するサンプルの2次元配列または行列と見なされる、または見なされ得る。配列内のサンプルは、ピクセル（ピクチャ要素の短い形）またはペルとも呼ばれ得る。配列またはピクチャの水平および垂直の方向（または軸）のサンプルの数は、ピクチャのサイズおよび/または解像度を定義する。色の表現のために、通常は3つの色成分が用いられる、すなわち、ピクチャは、3つのサンプル配列で表され得る、または3つのサンプル配列を含み得る。RGB形式または色空間では、ピクチャは、対応する赤、緑、および青のサンプルの配列を備える。しかしながら、ビデオ符号化では、各ピクセルは、通常、輝度/クロミナンス形式または色空間、例えば、YCbCrで表され、YCbCrは、Yによって示される輝度成分（場合によっては、Lが代わりに使用されることもある）と、CbおよびCrによって示される2つのクロミナンス成分とを含む。輝度（または略してルマ）成分Yは、明るさまたはグレースケールの強度（例えば、グレースケールピクチャの場合）を表し、2つのクロミナンス（または略してクロマ）成分CbおよびCrは、色度

10

20

30

40

50

または色情報成分を表す。したがって、YCbCr形式のピクチャは、輝度サンプル値（Y）の輝度サンプル配列ならびにクロミナンス値（CbおよびCr）の2つのクロミナンスサンプル配列を含む。RGB形式のピクチャは、YCbCr形式に転換または変換され得、その逆も同様であり、このプロセスはまた、色変換または色転換として知られている。ピクチャがモノクロである場合、ピクチャは、輝度サンプル配列のみを含み得る。

【0157】

ピクチャソース312は、例えば、ピクチャを取り込むためのカメラ、以前に取り込まれたもしくは生成されたピクチャを含むまたは記憶するメモリ、例えばピクチャメモリ、および／またはピクチャを取得または受信するための任意の種類のインターフェース（内部または外部）であってもよい。カメラは、例えば、ソースデバイスに統合されたローカルまたは統合カメラであってもよく、メモリは、例えばソースデバイスに統合されたローカルまたは統合メモリであってもよい。インターフェースは、例えば、外部ビデオソース、例えば、カメラのような外部ピクチャ取り込みデバイス、外部メモリ、または外部ピクチャ生成デバイス、例えば、外部コンピュータグラフィックプロセッサ、コンピュータ、もしくはサーバからピクチャを受信するための外部インターフェースであってもよい。インターフェースは、任意の種類のインターフェース、例えば、専用または標準のインターフェースプロトコルに準拠した有線または無線インターフェース、光インターフェースであり得る。ピクチャデータ313を取得するためのインターフェースは、通信インターフェース318と同じインターフェースまたはその一部であり得る。通信インターフェースは、イーサネット、WLAN、Bluetooth、LTE、または衛星もしくは光インターフェースなどの有線もしくは非有線インターフェースなどの任意のインターフェースであってもよい。送信は、ピアツーピアまたはブロードキャストまたはマルチキャストであってもよい。

【0158】

前処理ユニット314および前処理ユニット314によって実行される処理と区別して、ピクチャまたはピクチャデータ313は、未加工ピクチャまたは未加工ピクチャデータ313と呼ばれる場合もある。

【0159】

前処理ユニット314は、（未加工）ピクチャデータ313を受信し、前処理されたピクチャ315または前処理されたピクチャデータ315を取得するためにピクチャデータ313に対して前処理を実行するように構成される。前処理ユニット314によって実行される前処理は、例えば、トリミング、カラー形式転換（例えば、RGBからYCbCrへの）、色補正、またはノイズ除去を含み得る。

【0160】

エンコーダ100は、前処理されたピクチャデータ315を受信し、エンコードされたピクチャデータを提供するように構成される（さらなる詳細は、例えば図1に基づいて説明されている）。

【0161】

ソースデバイス310の通信インターフェース318は、記憶または直接再構築のために、エンコードされたピクチャデータを受信し、これを別のデバイス、例えば宛先デバイス320もしくは任意の他のデバイスに直接送信するか、またはデコーディングまたは記憶のために、エンコードされたデータ330を記憶する前におよび／もしくはエンコードされたデータ330を別のデバイス、例えば宛先デバイス320もしくは任意の他のデバイスに送信する前に、それぞれのために、エンコードされたピクチャデータを処理するように構成され得る。

【0162】

宛先デバイス320は、デコーダ200またはデコーディングユニット200を備え、追加的に、すなわち任意選択で、通信インターフェースまたは通信ユニット322、後処理ユニット326、および表示デバイス328を備え得る。

【0163】

宛先デバイス320の通信インターフェース322は、例えば、ソースデバイス310から直

10

20

30

40

50

接、または任意の他のソース、例えばメモリ、例えばエンコードされたピクチャデータのメモリから、エンコードされたピクチャデータまたはエンコードされたデータ330を受信するように構成される。

【0164】

通信インターフェース318および通信インターフェース322は、ソースデバイス310と宛先デバイス320との間の直接通信リンク、例えば、直接的な有線もしくは無線接続を介して、または任意の種類のネットワーク、例えば、有線（光、電力線、クーパー、同軸、もしくは任意の他の媒体に基づくものなど）もしくは無線ネットワークまたはこれらの任意の組み合わせ、または任意の種類のプライベートネットワークおよびパブリックネットワークもしくはこれらの任意の種類の組み合わせを介して、エンコードされたピクチャデータまたはエンコードされたデータ330をそれぞれ送信、受信するように構成され得る。

10

【0165】

通信インターフェース318は、例えば、通信リンクまたは通信ネットワークを介した送信のための適切な形式、例えばパケットに、エンコードされたピクチャデータをパッケージ化するように構成され得、データ損失保護およびデータ損失回復をさらに含み得る。

【0166】

通信インターフェース318の対応物を形成する通信インターフェース322は、例えば、エンコードされたピクチャデータを取得するために、エンコードされたデータ330をデパッケージ化するように構成され得、さらに、例えば誤り隠蔽を含むデータ損失保護およびデータ損失回復を実行するように構成され得る。

20

【0167】

通信インターフェース318と通信インターフェース322の両方は、ソースデバイス310から宛先デバイス320を指す、図14のエンコードされたピクチャデータ330の矢印によって示されているような単方向通信インターフェースまたは双方向通信インターフェースとして構成され得、例えばメッセージを送信および受信するように、例えば、接続をセットアップするように、ピクチャデータを含む失われたまたは遅延したデータを確認および／または再送信し、通信リンクおよび／またはデータ送信、例えばエンコードされたピクチャデータ送信に関連する任意の他の情報を交換するように構成され得る。

【0168】

デコーダ200は、エンコードされたピクチャデータを受信し、デコードされたピクチャデータまたはデコードされたピクチャを提供するように構成される（さらなる詳細は、例えば図2に基づいて説明されている）。

30

【0169】

宛先デバイス320のポストプロセッサ326は、後処理されたピクチャデータ327、例えば後処理されたピクチャ327を取得するために、デコードされたピクチャデータ、例えばデコードされたピクチャを後処理するように構成される。後処理ユニット326によって実行される後処理は、例えば、カラー形式転換（例えば、YCbCrからRGBへの）、色補正、トリミング、もしくはリサンプリング、または例えば、デコードされたピクチャデータを、例えば表示デバイス328による表示のために準備するための任意の他の処理を含み得る。

【0170】

宛先デバイス320の表示デバイス328は、ピクチャを例えばユーザまたはビューアに表示するために、後処理されたピクチャデータ327を受信するように構成される。表示デバイス328は、再構築されたピクチャを表すための任意の種類のディスプレイ、例えば、統合もしくは外部ディスプレイもしくはモニターであってもよいし、またはこれを備えてもよい。ディスプレイは、例えば、陰極線管（CRT）、液晶ディスプレイ（LCD）、プラズマディスプレイ、有機発光ダイオード（OLED）ディスプレイ、またはビーマ、ホログラム、もしくは3D／VRグラスを含む任意の種類の他のディスプレイを備えてもよい。

40

【0171】

図14は、ソースデバイス310および宛先デバイス320を別個のデバイスとして示しているが、デバイスの実施形態はまた、両方または両方の機能、ソースデバイス310または対

50

応する機能ならびに宛先デバイス320または対応する機能を含み得る。このような実施形態では、ソースデバイス310または対応する機能ならびに宛先デバイス320または対応する機能は、同じハードウェアおよび／またはソフトウェアを使用して、または別個のハードウェアおよび／またはソフトウェアまたはこれらの任意の組み合わせによって実施され得る。

【0172】

説明に基づいて当業者には明らかなように、図14に示されているようなソースデバイス310および／または宛先デバイス320内の異なるユニットまたは機能の機能の存在および（正確な）分割は、実際のデバイスおよび用途に応じて変更され得る。

【0173】

したがって、図14に示されているようなソースデバイス310および宛先デバイス320は、本発明の単なる例示的な実施形態であり、本発明の実施形態は、図14に示されているものに限定されない。

【0174】

ソースデバイス310および宛先デバイス320は、例えば、ノートブックもしくはラップトップコンピュータ、携帯電話、スマートフォン、タブレット、もしくはタブレットコンピュータ、カメラ、デスクトップコンピュータ、セットトップボックス、テレビ、表示デバイス、デジタルメディアプレーヤ、ビデオゲーミングコンソール、ビデオストリーミングデバイス、またはブロードキャストレシーバデバイスなど、任意の種類のハンドヘルドまたは固定デバイスを含む、広範囲のデバイスのいずれかを含み得、オペレーティングシステムをまったく使用し得ないか、または任意の種類のオペレーティングシステムを使用し得る。

【0175】

図15は、本開示の一実施形態によるビデオ符号化デバイス1000の概略図である。ビデオ符号化デバイス1000は、例えばエンコーダまたはデコーダとして、本明細書で説明されているような開示された実施形態を実施するのに適している。ビデオ符号化デバイス1000は、データを受信するための入力ポート1010および受信機ユニット（Rx）1020と、データを処理するためのプロセッサ、論理ユニット、または中央処理装置（CPU）1030と、データを送信するための送信機ユニット（Tx）1040および出力ポート1050と、データを記憶するためのメモリ1060とを備える。ビデオ符号化デバイス1000はまた、光信号または電気信号の出力または入力のために入力ポート1010、受信機ユニット1020、送信機ユニット1040、および出力ポート1050に結合された光－電気（OE）コンポーネントおよび電気－光（EO）コンポーネントを備え得る。

【0176】

本開示は、装置で実施され得る。このような装置は、ソフトウェアとハードウェアの組み合わせであり得る。例えば、イントラ予測およびデブロッキングフィルタリングは、汎用プロセッサ、またはデジタル信号プロセッサ（DSP）、またはフィールドプログラマブルゲートアレイ（FPGA）などのチップによって実行され得る。しかしながら、本発明は、プログラマブルハードウェア上での実施に限定されない。それは、特定用途向け集積回路（ASIC）上でまたは上で言及されたハードウェアコンポーネントの組み合わせによって実施され得る。

【0177】

本開示は、処理回路によって実行されると、処理回路に、画像を分割、エンコーディング、およびデコーディングするための開示された方法のいずれかを実行させる命令を記憶したコンピュータ可読媒体をさらに提供する。コンピュータ可読媒体は、DVD、CD、USB（フラッシュ）ドライブ、ハードディスク、ネットワーク経由で利用可能なサーバストレージなど、プログラムが記憶された任意の媒体であってもよい。

【0178】

エンコーダおよび／またはデコーダは、TVセット、セットトップボックス、PC、タブレット、またはスマートフォンなどを含む様々なデバイスにおいて実施されてもよい。そ

れは、方法ステップを実施するソフトウェア、アプリであってもよい。

【0179】

要約すると、本開示は、画像を符号化ユニットに分割するための装置および方法を提供する。画像は、階層的に区分化された符号化ツリーユニット（CTU）に分割される。階層的区分化は、二分木または四分木の分割などのマルチタイプ区分化を含む。完全に画像内にあるCTUおよび境界上のCTUに関して、それぞれのマルチタイプ区分深度が選択される。本開示は、画像の境界部分におけるマルチタイプ区分化の柔軟性を提供する。

【符号の説明】

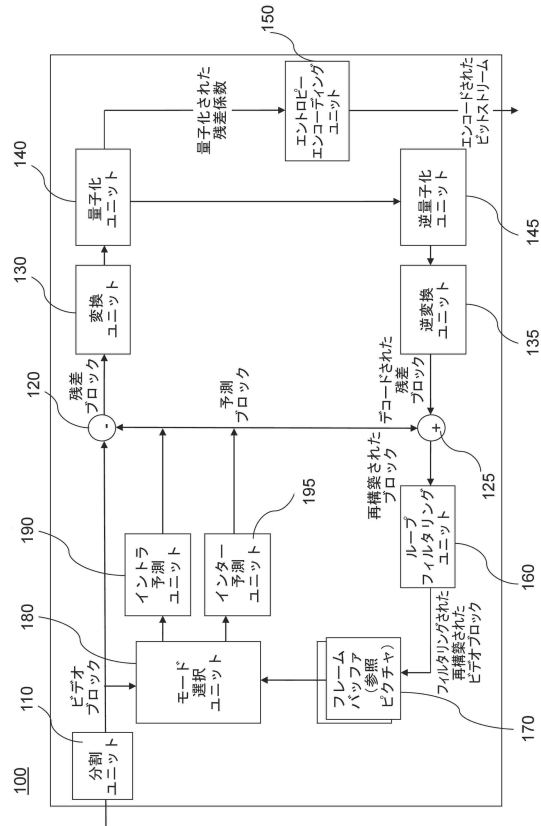
【0180】

100	エンコーダ、ビデオエンコーディング装置	10
110	分割ユニット	
120	残差ブロック	
125	再構築器	
130	変換ユニット	
135	逆変換ユニット	
140	量子化ユニット	
145	逆量子化ユニット	
150	エントロピーエンコーディングユニット、エントロピー符号化ユニット	
160	ループフィルタリングユニット	
170	フレームバッファ、参照ピクチャバッファ	20
180	モード選択ユニット	
190	イントラ予測ユニット	
195	インター予測ユニット	
200	ビデオデコーダ、デコーディング装置	
225	再構築器	
230	逆変換ユニット	
240	逆量子化ユニット	
250	ビットストリーム解析、エントロピーデコーディング、および分割ユニット	
260	ループフィルタリングユニット	
270	参照ピクチャバッファ	30
290	イントラ予測ユニット	
295	インター予測ユニット	
300	ピクチャ符号化システム	
310	ソースデバイス	
312	ピクチャソース	
313	未加工ピクチャデータ	
314	プリプロセッサ、ピクチャ前処理ユニット	
315	前処理されたピクチャデータ	
318	通信インターフェース、通信ユニット	
320	宛先デバイス	40
322	通信インターフェース、通信ユニット	
326	ポストプロセッサ、後処理ユニット	
327	後処理されたピクチャ	
328	表示デバイス	
330	エンコードされたデータ	
800	画像を符号化ユニットに分割するための装置	
810	画像分割、フレーム分割、機能ユニット	
820	境界区分化	
900	下部ピクチャ境界	
910	境界部分	50

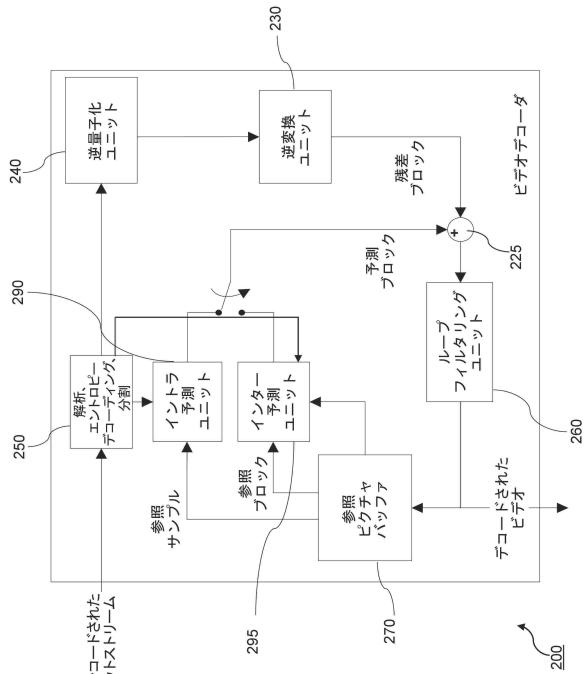
- 920 部分
- 950 部分
- 1000 ビデオ符号化デバイス
- 1010 入力ポート
- 1020 受信機ユニット
- 1030 プロセッサ
- 1040 送信機ユニット
- 1050 出力ポート
- 1060 メモリ
- 1070 符号化モジュール

10

【図面】
【図 1】



【図 2】



20

30

40

50

【図 7】

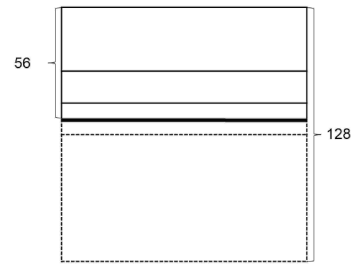
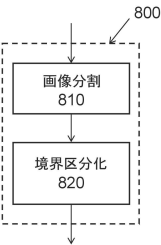


Fig. 7

【図 8】



10

【図 9】

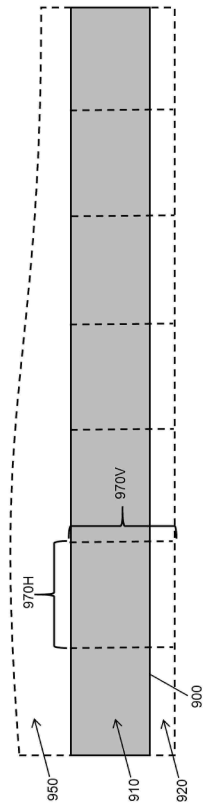
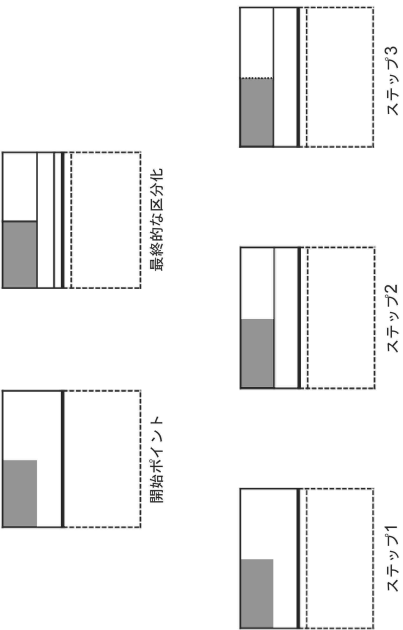


Fig. 9

【図 10】



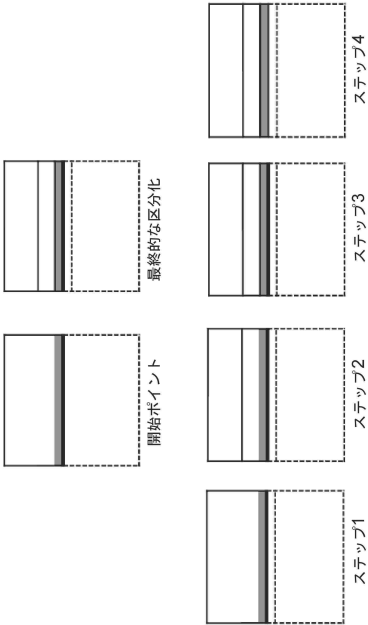
20

30

40

50

【図 1 1】



【図 1 2】

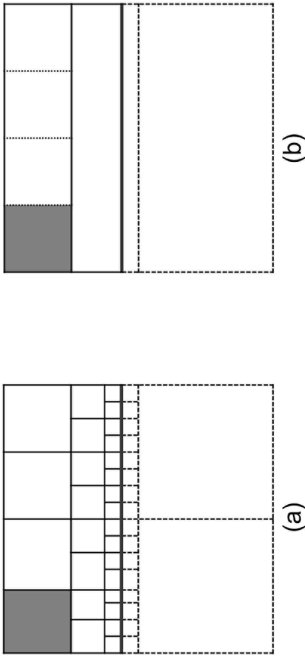
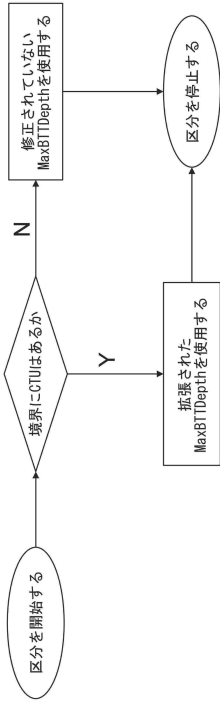
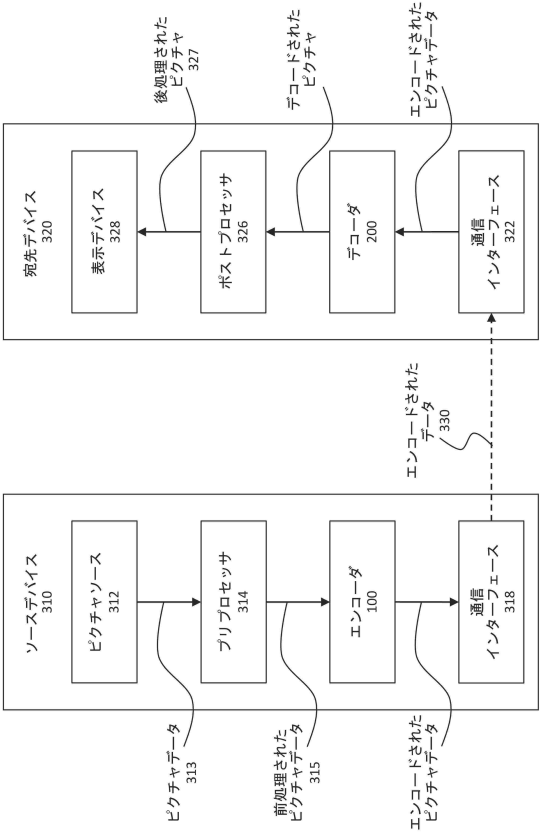


Fig. 12

【図 1 3】



【図 1 4】



300

10

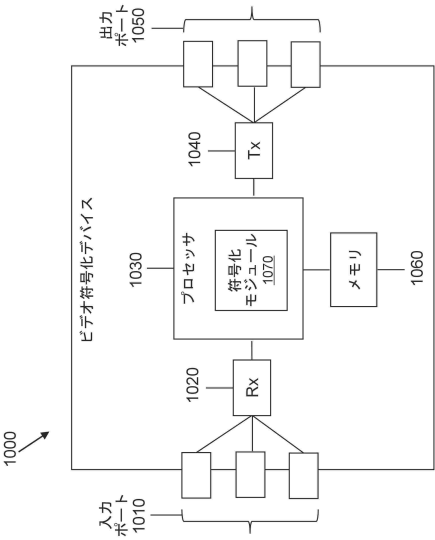
20

30

40

50

【図 15】



10

20

30

40

50

フロントページの続き

- クノロジーズ・デュッセルドルフ・ゲーエムベーハー
- (72)発明者 ジジエ・ジャオ
ドイツ・８０９９２・ミュンヘン・リースシュトラッセ・２５・ホアウェイ・テクノロジーズ・デュッセルドルフ・ゲーエムベーハー
- (72)発明者 セミフ・エセンリク
ドイツ・８０９９２・ミュンヘン・リースシュトラッセ・２５・ホアウェイ・テクノロジーズ・デュッセルドルフ・ゲーエムベーハー
- (72)発明者 アナンド・メヘル・コトラ
ドイツ・８０９９２・ミュンヘン・リースシュトラッセ・２５・ホアウェイ・テクノロジーズ・デュッセルドルフ・ゲーエムベーハー
- (72)発明者 ジエンレ・チェン
アメリカ合衆国・カリフォルニア・９５０５０・サンタ・クララ・セントラル・エクスプレスウェイ・２３３０
- 審査官 岩井 健二
- (56)参考文献 米国特許出願公開第２０１７／０２７２７８２（ＵＳ，Ａ１）
MA, Jackie et al., Quadtree plus binary tree with shifting (including software), Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 10th Meeting; JVET-J0035 (version 4), ITU-T, 2018年04月, pp.1-22, <URL:http://phenix.it-sudparis.eu/jvet/doc_end_user/documents/10_San%20Diego/wg11/JVET-J0035-v5.zip
WU, Feng et al., Description of SDR video coding technology proposal by University of Science and Technology of China, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 10th Meeting; San, JVET-J0032 (version 4), ITU-T, 2018年04月, pp.2-9,17, <URL:http://phenix.it-sudparis.eu/jvet/doc_end_user/documents/10_San%20Diego/wg11/JVET-J0032-v4.zip
- (58)調査した分野 (Int.Cl., D B名)
H 0 4 N 1 9 / 0 0 - 1 9 / 9 8