

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号  
特許第7510412号  
(P7510412)

(45)発行日 令和6年7月3日(2024.7.3)

(24)登録日 令和6年6月25日(2024.6.25)

(51)国際特許分類 F I  
H 0 4 N 19/70 (2014.01) H 0 4 N 19/70  
H 0 4 N 19/96 (2014.01) H 0 4 N 19/96

請求項の数 20 (全38頁)

(21)出願番号	特願2021-521515(P2021-521515)	(73)特許権者	518338149
(86)(22)出願日	令和1年10月21日(2019.10.21)		インターデジタル ヴイシー ホールディ ングス, インコーポレイテッド
(65)公表番号	特表2022-505470(P2022-505470 A)		アメリカ合衆国, デラウェア州 1 9 8 0 9, ウィルミントン, ベルビュー パ ークウェイ 2 0 0, スイート 3 0 0
(43)公表日	令和4年1月14日(2022.1.14)	(74)代理人	100079108
(86)国際出願番号	PCT/US2019/057149		弁理士 稲葉 良幸
(87)国際公開番号	WO2020/096755	(74)代理人	100109346
(87)国際公開日	令和2年5月14日(2020.5.14)		弁理士 大貫 敏史
審査請求日	令和4年10月12日(2022.10.12)	(74)代理人	100117189
(31)優先権主張番号	18306469.0		弁理士 江口 昭彦
(32)優先日	平成30年11月8日(2018.11.8)	(74)代理人	100134120
(33)優先権主張国・地域又は機関	欧州特許庁(EP)		弁理士 内藤 和彦
(31)優先権主張番号	18306824.6	(74)代理人	100108213
(32)優先日	平成30年12月21日(2018.12.21)		
	最終頁に続く		最終頁に続く

(54)【発明の名称】 ブロックの面に基づくビデオの符号化又は復号のための量子化

(57)【特許請求の範囲】

【請求項1】

ビデオをコード化する方法であって、

画像のブロックがサブブロックに分割されたことを決定することと、  
サブブロックごとに、当該サブブロックに再分割レベルカウンタを関連付けることであって、前記再分割レベルカウンタは、前記サブブロックのサイズを表し、サブブロックへの分割の分割の種類に基づいて複数のインクリメント値から選択されるインクリメント値を有し、前記インクリメント値は、クワッド分割の全てのサブブロックについて2、バイナリ分割の全てのサブブロックについて1、ターナリ分割の横のサブブロックについて2、及びターナリ分割の中央のサブブロックについて1に等しい、ことと、

前記再分割レベルカウンタに基づいて、前記サブブロックの量子化パラメータを決定することと、

前記決定された量子化パラメータに基づいて、前記サブブロックをコード化することとを含む、方法。

【請求項2】

ビデオを復号する方法であって、

画像のブロックがサブブロックに分割されたことを決定することと、  
サブブロックごとに、当該サブブロックに再分割レベルカウンタを関連付けることであって、前記再分割レベルカウンタは、前記サブブロックのサイズを表し、サブブロックへの分割の分割の種類に基づいて複数のインクリメント値から選択されるインクリメント値を

有し、前記インクリメント値は、クワッド分割の全てのサブブロックについて2、バイナリ分割の全てのサブブロックについて1、ターナリ分割の横のサブブロックについて2、及びターナリ分割の中央のサブブロックについて1に等しい、ことと、

前記再分割レベルカウンタに基づいて、前記サブブロックの量子化パラメータを決定することと、

前記決定された量子化パラメータに基づいて、前記サブブロックを復号することとを含む、方法。

【請求項3】

符号器を含む、ビデオを符号化する装置であって、前記符号器が、  
 画像のブロックがサブブロックに分割されたことを決定することと、  
 サブブロックごとに、当該サブブロックに再分割レベルカウンタを関連付けることであって、前記再分割レベルカウンタは、前記サブブロックのサイズを表し、サブブロックへの分割の種類に基づいて複数のインクリメント値から選択されるインクリメント値を有し、前記インクリメント値は、クワッド分割の全てのサブブロックについて2、バイナリ分割の全てのサブブロックについて1、ターナリ分割の横のサブブロックについて2、及びターナリ分割の中央のサブブロックについて1に等しい、ことと、

10

前記再分割レベルカウンタに基づいて、前記サブブロックの量子化パラメータを決定することと、

前記決定された量子化パラメータに基づいて、前記サブブロックをコード化することとを行うように構成された、装置。

20

【請求項4】

復号器を含む、ビデオを復号する装置であって、前記復号器が、  
 画像のブロックがサブブロックに分割されたことを決定することと、  
 サブブロックごとに、当該サブブロックに再分割レベルカウンタを関連付けることであって、前記再分割レベルカウンタは、前記サブブロックのサイズを表し、サブブロックへの分割の種類に基づいて複数のインクリメント値から選択されるインクリメント値を有し、前記インクリメント値は、クワッド分割の全てのサブブロックについて2、バイナリ分割の全てのサブブロックについて1、ターナリ分割の横のサブブロックについて2、及びターナリ分割の中央のサブブロックについて1に等しい、ことと、

前記再分割レベルカウンタに基づいて、前記サブブロックの量子化パラメータを決定することと、

30

前記決定された量子化パラメータに基づいて、前記サブブロックを復号することとを行うように構成された、装置。

【請求項5】

サブブロックの前記再分割レベルカウンタが再分割の最大値以下である間、前記分割の種類に従って新たな量子化グループを有効にすること  
 を更に含む、請求項2に記載の方法。

【請求項6】

新たな量子化グループが有効にされるとき、前記ブロックの位置を記憶すること  
 を更に含む、請求項5に記載の方法。

40

【請求項7】

画像のブロックがターナリ分割に従ってサブブロックに分割され、且つ最も小さいサブブロックの前記再分割レベルカウンタが再分割の最大値を上回るとき、新たな量子化グループを無効にすること  
 を更に含む、請求項5又は6に記載の方法。

【請求項8】

量子化は、ルマ成分又はクロマ成分に適用される、請求項1に記載の方法。

【請求項9】

前記復号器は、サブブロックの前記再分割レベルカウンタが再分割の最大値以下である間、前記分割の種類に従って新たな量子化グループを有効にすること

50

を行うようにさらに構成されている、請求項 4 に記載の装置。

【請求項 10】

前記復号器は、新たな量子化グループが有効にされるとき、前記ブロックの位置を記憶すること

を行うようにさらに構成されている、請求項 9 に記載の装置。

【請求項 11】

前記復号器は、画像のブロックがターナリ分割に従ってサブブロックに分割され、且つ最も小さいサブブロックの前記再分割レベルカウンタが再分割の最大値を上回るとき、新たな量子化グループを無効にすること

を行うようにさらに構成されている、請求項 9 又は 10 に記載の装置。

10

【請求項 12】

量子化は、ルマ成分又はクロマ成分に適用される、請求項 4 に記載の装置。

【請求項 13】

ビデオをコード化する方法であって、

画像のブロックがサブブロックに分割されたことを決定することと、

サブブロックごとに、当該サブブロックに再分割レベルカウンタを関連付けることであって、前記再分割レベルカウンタは、前記サブブロックのサイズを表し、サブブロックへの分割の種類に基づいて複数のインクリメント値から選択されるインクリメント値を有し、前記インクリメント値は、クワッド分割の全てのサブブロックについて 4、バイナリ分割の全てのサブブロックについて 2、ターナリ分割の横のサブブロックについて 4、及び中央のサブブロックについて 2、非対称 1 / 4 バイナリ分割の最小のサブブロックについて 4、及び最大のサブブロックについて 1、並びに非対称 1 / 3 バイナリ分割の最小のサブブロックについて 3、及び最大のサブブロックについて 1 に等しい、ことと、前記再分割レベルカウンタに基づいて、前記サブブロックの量子化パラメータを決定することと、

20

前記決定された量子化パラメータに基づいて、前記サブブロックをコード化することとを含む、方法。

【請求項 14】

ビデオを復号する方法であって、

画像のブロックがサブブロックに分割されたことを決定することと、

サブブロックごとに、当該サブブロックに再分割レベルカウンタを関連付けることであって、前記再分割レベルカウンタは、前記サブブロックのサイズを表し、サブブロックへの分割の種類に基づいて複数のインクリメント値から選択されるインクリメント値を有し、前記インクリメント値は、クワッド分割の全てのサブブロックについて 4、バイナリ分割の全てのサブブロックについて 2、ターナリ分割の横のサブブロックについて 4、及び中央のサブブロックについて 2、非対称 1 / 4 バイナリ分割の最小のサブブロックについて 4、及び最大のサブブロックについて 1、並びに非対称 1 / 3 バイナリ分割の最小のサブブロックについて 3、及び最大のサブブロックについて 1 に等しい、ことと、前記再分割レベルカウンタに基づいて、前記サブブロックの量子化パラメータを決定することと、

30

前記決定された量子化パラメータに基づいて、前記サブブロックを復号することとを含む、方法。

40

【請求項 15】

符号器を含む、ビデオを符号化する装置であって、前記符号器が、

画像のブロックがサブブロックに分割されたことを決定することと、

サブブロックごとに、当該サブブロックに再分割レベルカウンタを関連付けることであって、前記再分割レベルカウンタは、前記サブブロックのサイズを表し、サブブロックへの分割の種類に基づいて複数のインクリメント値から選択されるインクリメント値を有し、前記インクリメント値は、クワッド分割の全てのサブブロックについて 4、バイナリ分割の全てのサブブロックについて 2、ターナリ分割の横のサブブロックについて 4、

50

及び中央のサブブロックについて2、非対称1/4バイナリ分割の最小のサブブロックについて4、及び最大のサブブロックについて1、並びに非対称1/3バイナリ分割の最小のサブブロックについて3、及び最大のサブブロックについて1に等しい、ことと、  
前記再分割レベルカウンタに基づいて、前記サブブロックの量子化パラメータを決定することと、

前記決定された量子化パラメータに基づいて、前記サブブロックをコード化することとを行うように構成された、装置。

【請求項16】

復号器を含む、ビデオを復号する装置であって、前記復号器が、

画像のブロックがサブブロックに分割されたことを決定することと、

サブブロックごとに、当該サブブロックに再分割レベルカウンタを関連付けることであって、前記再分割レベルカウンタは、前記サブブロックのサイズを表し、サブブロックへの分割の種類に基づいて複数のインクリメント値から選択されるインクリメント値を有し、前記インクリメント値は、クワッド分割の全てのサブブロックについて4、バイナリ分割の全てのサブブロックについて2、ターナリ分割の横のサブブロックについて4、及び中央のサブブロックについて2、非対称1/4バイナリ分割の最小のサブブロックについて4、及び最大のサブブロックについて1、並びに非対称1/3バイナリ分割の最小のサブブロックについて3、及び最大のサブブロックについて1に等しい、ことと、  
前記再分割レベルカウンタに基づいて、前記サブブロックの量子化パラメータを決定することと、

前記決定された量子化パラメータに基づいて、前記サブブロックを復号することとを行うように構成された、装置。

【請求項17】

請求項1又は1.3に記載の方法のステップを実施する、プロセッサによって実行可能なプログラムコード命令を含むコンピュータプログラム。

【請求項18】

請求項2又は1.4に記載の方法のステップを実施する、プロセッサによって実行可能なプログラムコード命令を含むコンピュータプログラム。

【請求項19】

プロセッサによって実行されるとき、請求項1又は1.3に記載の方法を装置が実施することを可能にする命令を記憶した非一時的コンピュータ可読記憶媒体。

【請求項20】

プロセッサによって実行されるとき、請求項2又は1.4に記載の方法を装置が実施することを可能にする命令を記憶した非一時的コンピュータ可読記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、ビデオ圧縮の分野におけるものであり、少なくとも1つの実施形態は、より詳細には、ブロックの面(surface)に基づいて量子化パラメータを決定することに関する。

【背景技術】

【0002】

高い圧縮効率を実現するために、画像及びビデオのコード化方式は、通常、予測及び変換を使用してビデオコンテンツ内の空間的及び時間的な冗長性を活用する。概して、フレーム内又はフレーム間の相関を利用するためにイントラ予測又はインター予測が使用され、その後、予測誤差又は予測残差として示されることが多い、元のブロックと予測ブロックとの差が変換され、量子化され、エントロピーコード化される。ビデオを再構築するために、エントロピーコード化、量子化、変換及び予測に対応する逆のプロセスによって圧縮データが復号される。

【発明の概要】

10

20

30

40

50

## 【課題を解決するための手段】

## 【0003】

少なくとも1つの実施形態の第1の態様によれば、ビデオをコード化する方法は、画像のブロックがサブブロックに分割されるとき、サブブロックの面を表す値を関連付けること、及びサブブロックの面を表す値に基づいて、サブブロックの量子化パラメータを決定することを含む。

## 【0004】

少なくとも1つの実施形態の第2の態様によれば、ビデオを復号する方法は、画像のブロックがサブブロックに分割されるとき、サブブロックの面を表す値を関連付けること、及びサブブロックの面を表す値に基づいて、サブブロックの量子化パラメータを決定することを含む。

10

## 【0005】

少なくとも1つの実施形態の第3の態様によれば、ピクチャデータを符号化するビデオ符号器を含む機器は、画像のブロックがサブブロックに分割されるとき、サブブロックの面を表す値を関連付けること、及びサブブロックの面を表す値に基づいて、サブブロックの量子化パラメータを決定することを含む。

## 【0006】

少なくとも1つの実施形態の第4の態様によれば、ピクチャデータを復号するビデオ復号器を含む機器は、画像のブロックがサブブロックに分割されるとき、サブブロックの面を表す値を関連付けること、及びサブブロックの面を表す値に基づいて、サブブロックの量子化パラメータを決定することを含む。

20

## 【0007】

第1、第2、第3又は第4の態様の変形実施形態によれば、サブブロックの面を表す値は、画像のブロックがサブブロックに分割されるとき、サブブロックの分割の種類を表す値だけインクリメントされる再分割レベルカウンタである。第1、第2、第3又は第4の態様の更なる変形実施形態によれば、サブブロックの再分割レベルカウンタが再分割の最大値以下である間、分割の種類に従って新たな量子化グループが有効にされる。第1、第2、第3又は第4の態様の更なる変形実施形態によれば、画像のブロックがターナリ分割に従ってサブブロックに分割され、且つ最も小さいサブブロックの再分割レベルカウンタが再分割の最大値を上回るとき、新たな量子化グループが無効にされる。

30

## 【0008】

第1、第2、第3若しくは第4の態様の変形実施形態又はこれらの実施形態の変形形態によれば、インクリメント値は、クワッド分割の全てのサブブロックについて、2、バイナリ分割の全てのサブブロックについて、1、ターナリ分割の横のサブブロックについて、2、及びターナリ分割の中央のサブブロックについて、1に等しい。第1、第2、第3若しくは第4の態様の別の変形実施形態又はこれらの実施形態の変形形態によれば、インクリメント値は、クワッド分割の全てのサブブロックについて、4、バイナリ分割の全てのサブブロックについて、2、ターナリ分割の横のサブブロックについて、4、及び中央のサブブロックについて、2、非対称1/4バイナリ分割の最小のサブブロックについて、4、及び最大のサブブロックについて、1、並びに非対称1/3バイナリ分割の最小のサブブロックについて、3、最大のサブブロックについて、1に等しい。

40

## 【0009】

少なくとも1つの実施形態の第5の態様によれば、記憶媒体は、サブブロックの面を表す値を表す少なくとも構文データ要素を含むビデオビットストリームデータを記憶媒体が記憶し、サブブロックは、ビデオの画像のブロックを分割することの結果である。第5の実施形態の変形実施形態によれば、サブブロックの面を表す値は、画像のブロックがサブブロックに分割されるとき、サブブロックの分割の種類を表す値だけインクリメントされる再分割レベルカウンタである。第5の実施形態の更なる変形実施形態によれば、インクリメント値は、クワッド分割の全てのサブブロックについて、2、バイナリ分割の全てのサブブロックについて、1、ターナリ分割の横のサブブロックについて、2、及びターナ

50

リ分割の中央のサブブロックについて、1に等しい。

【0010】

少なくとも1つの実施形態の第6の態様によれば、プロセッサによって実行可能なプログラムコード命令を含むコンピュータプログラムが示され、コンピュータプログラムは、少なくとも第1の態様又は第2の態様による方法のステップを実施する。

【0011】

少なくとも1つの実施形態の第7の態様によれば、非一時的コンピュータ可読媒体上に記憶され、且つプロセッサによって実行可能なプログラムコード命令を含むコンピュータプログラム製品が示され、コンピュータプログラム製品は、少なくとも第1の態様又は第2の態様による方法のステップを実施する。

10

【図面の簡単な説明】

【0012】

【図1】ビデオ符号器100の一例のブロック図を示す。

【図2】ビデオ復号器200の一例のブロック図を示す。

【図3】様々な態様及び実施形態が実装されるシステムの一例のブロック図を示す。

【図4】圧縮領域内のコード化ツリー単位及びコード化ツリーの一例を示す。

【図5】CTUをコード化単位、予測単位及び変換単位に分割する一例を示す。

【図6】例えば、VVCにあるような様々な分割の種類を示す。

【図7A】ブロックサイズに対する繰り返し分割の影響を示す。

【図7B】ブロックサイズに対する繰り返し分割の影響を示す。

20

【図8】異なる分割レベルにおける同じサイズの様々なブロックを示す。

【図9】HEVC仕様における量子化グループの使用を示す。

【図10】再分割レベルカウンタsubdivの使用に関する少なくとも1つの実施形態を示す。

【図11A】一実施形態による、分割の種類とsubdivのインクリメントとの間の関係の一例を示す。

【図11B】一実施形態による、分割の種類とsubdivのインクリメントとの間の関係の一例を示す。

【図11C】一実施形態による、分割の種類とsubdivのインクリメントとの間の関係の一例を示す。

30

【図12】一実施形態による、subdivプロセスのブロック図の一例を示す。

【図13A】図6及び図7の例にプロセスを適用したときのsubdivカウンタの値の一例を示す。

【図13B】図6及び図7の例にプロセスを適用したときのsubdivカウンタの値の一例を示す。

【図13C】図6及び図7の例にプロセスを適用したときのsubdivカウンタの値の一例を示す。

【図14A】異なるタイプの非対称バイナリ分割を示す。

【図14B】異なるタイプの非対称バイナリ分割を示す。

【図14C】T分割の結果の一例を示す。

40

【図15A】AB分割に関する第1の変形実施形態による、分割の種類とsubdivのインクリメントとの間の関係の一例を示す。

【図15B】AB分割に関する第1の変形実施形態による、分割の種類とsubdivのインクリメントとの間の関係の一例を示す。

【図16A】ターナリ分割に相当する連続した2回の非対称バイナリ分割にプロセスを適用したときのsubdivカウンタの値の一例を示す。

【図16B】ターナリ分割に相当する連続した2回の非対称バイナリ分割にプロセスを適用したときのsubdivカウンタの値の一例を示す。

【図16C】ターナリ分割に相当する連続した2回の非対称バイナリ分割にプロセスを適用したときのsubdivカウンタの値の一例を示す。

50

【図 1 7】 A B 分割に関する第 1 の変形形態を用いた、相対する方向の連続した 2 回の非対称 1 / 4 バイナリ分割にプロセスを適用したときの subdiv カウンタの値の一例を示す。

【図 1 8 A】 A B 分割に関する第 2 の変形実施形態による、分割の種類と subdiv のインクリメントとの間の関係の一例を示す。

【図 1 8 B】 A B 分割に関する第 2 の変形実施形態による、分割の種類と subdiv のインクリメントとの間の関係の一例を示す。

【図 1 8 C】 A B 分割に関する第 2 の変形実施形態による、分割の種類と subdiv のインクリメントとの間の関係の一例を示す。

【図 1 8 D】 A B 分割に関する第 2 の変形実施形態による、分割の種類と subdiv のインクリメントとの間の関係の一例を示す。

10

【図 1 8 E】 A B 分割に関する第 2 の変形実施形態による、分割の種類と subdiv のインクリメントとの間の関係の一例を示す。

【図 1 9】 T 分割の場合の Q G 開始の一例を示す。

【図 2 0】 Q G の問題を訂正する一実施形態による、 T 分割の場合の Q G の画定の一例を示す。

【図 2 1】 Q G 識別プロセスの実施形態の一例のブロック図を示す。

【図 2 2】 Q G 識別プロセスの実施形態の別の例のブロック図を示す。

【図 2 3】 図 1 9 の状況におけるかかる実施形態の結果を示す。

【図 2 4】 再分割レベルカウンタを使用する実施形態による、ピクチャパラメータセット ( P P S ) の構文の一例を表すテーブルを示す。

20

【図 2 5】 主な実施形態を実装するための構文の一例を表すテーブルを示す。

【図 2 6】 主な実施形態を実装するための構文の一例を表すテーブルを示す。

【図 2 7】 主な実施形態を実装するための構文の一例を表すテーブルを示す。

【図 2 8】 主な実施形態を実装するための構文の一例を表すテーブルを示す。

【図 2 9】 主な実施形態を実装し、 Q G 開始検出を処理するための構文の一例を表すテーブルを示す。

【図 3 0】 主な実施形態を実装し、 Q G 開始検出を処理するための構文の一例を表すテーブルを示す。

【図 3 1】 主な実施形態を実装し、 Q G 開始検出を処理するための構文の一例を表すテーブルを示す。

30

【図 3 2】 主な実施形態を実装し、クロマ関連要素に関する Q G 開始検出を処理するための構文の一例を表すテーブルを示す。

【図 3 3】 主な実施形態を実装し、クロマ関連要素に関する Q G 開始検出を処理するための構文の一例を表すテーブルを示す。

【図 3 4】 主な実施形態を実装し、クロマ関連要素に関する Q G 開始検出を処理するための構文の一例を表すテーブルを示す。

【図 3 5】 主な実施形態を実装し、クロマ関連要素に関する Q G 開始検出を処理するための構文の一例を表すテーブルを示す。

【発明を実施するための形態】

【 0 0 1 3 】

40

少なくとも 1 つの実施形態では、ビデオの符号化又は復号が様々なサイズの複数のブロックに画像を分割し、再分割レベルカウンタ ( 以下では「subdiv」と呼ぶ ) は、パーティション及びブロックのそれぞれに関連付けられる。分割は、再帰的であり得、そのため、ブロックは ( サブブロックと呼ばれる ) 複数のブロック等に更に分割され得る。ブロック又はサブブロックに関するこのカウンタの値は、ブロック又はサブブロックのサイズを表し、ブロック又はサブブロックの量子化パラメータ及びブロックが属する量子化グループを決定するために使用される。その結果、同様のサイズのブロックに同じ量子化パラメータが使用され、従って一貫した符号化 / 復号を提供する。

【 0 0 1 4 】

説明を明確にするために、以下の説明は、例えば、Joint Video Experts Team ( J V

50

ET)によって策定されている新たな規格である高効率ビデオコーディング(HEVC)又は多用途ビデオコーディング(VVC)等のビデオ圧縮技術を含む実施形態に関して態様を説明する。但し、記載する態様は、他のビデオ処理技術及び規格にも適用可能である。

#### 【0015】

本願は、ツール、特徴、実施形態、モデル、手法等を含む様々な態様を記載する。これらの態様の多くは、特定の記載されており、少なくとも個々の特性を示すために限定的であるように思われ得る方法で多くの場合に説明されている。しかし、それは、説明を明瞭にすることを目的としており、それらの態様の応用又は範囲を限定するものではない。実際、様々な態様の全てを組み合わせ交換して更なる態様をもたらすことができる。更に、態様は、先の出願に記載の態様と組み合わせ交換することもできる。本願に記載し、本願で予期する態様は、多くの異なる形態で実装することができる。以下の図1、図2及び図3は、一部の実施形態を示すが、他の実施形態も予期され、図1、図2及び図3の解説は、実装形態の範囲を限定するものではない。態様の少なくとも1つは、概して、ビデオを符号化し復号することに関し、少なくとも1つの他の態様は、概して、生成又は符号化されたビットストリームを伝送することに関する。これらの及び他の態様は、方法、機器、記載する方法の何れかに従ってビデオデータを符号化又は復号するための命令を記憶しているコンピュータ可読記憶媒体及び/又は記載する方法の何れかに従って生成されるビットストリームを記憶しているコンピュータ可読記憶媒体として実装することができる。

10

#### 【0016】

本願では、「再構築する」という用語と、「復号する」という用語とを区別なく使用する場合があり、「ピクセル」という用語と、「サンプル」という用語とを区別なく使用する場合があり、「画像」、「ピクチャ」及び「フレーム」という用語を区別なく使用する場合がある。必ずではないが、通常、「再構築する」という用語は、符号器側で使用されるのに対し、「復号する」は、復号器側で使用される。

20

#### 【0017】

本明細書では、様々な方法を記載し、方法のそれぞれは、記載する方法を実現するための1つ又は複数のステップ又はアクションを含む。方法が適切に動作するのにステップ又はアクションの特定の順序が要求されない限り、特定のステップ及び/又はアクションの順序及び/又は使用は、修正するか又は組み合わせることができる。

#### 【0018】

本願に記載の様々な方法及び他の態様を使用して、モジュール、例えば図1に示すビデオ符号器(100)の量子化モジュール(130)及び/又は図2に示すビデオ復号器(200)の逆量子化モジュール(240)を修正することができる。更に、本願の態様は、VVC又はHEVCに限定されず、例えば既存の又は将来開発される他の規格及び勧告並びにそのような任意の規格及び勧告(VVC及びHEVCを含む)の拡張に適用することができる。別段の定めがない限り又は技術的に除外されない限り、本願に記載の態様は、個別に又は組み合わせで使用することができる。

30

#### 【0019】

本願では、例えば、ブロックサイズに関する様々な数値を使用する。具体的な値は、例示目的であり、記載する態様は、それらの具体的な値に限定されない。

40

#### 【0020】

図1は、ビデオ符号器100の一例のブロック図を示す。この符号器100の変形形態が考えられるが、符号器100は、予期される全ての変形形態を記載することなしに、明瞭にすることを目的に以下に記載される。例えば、図1の符号器は、HEVC規格に改善が加えられたHEVC符号器又はJVTによって開発されているJEM(Joint Exploration Model)符号器等のHEVCと同様の技術を使用する符号器であり得る。

#### 【0021】

ビデオシーケンスは、符号化される前に、例えば入力カラーピクチャに色変換(例えば、RGB4:4:4からYCbCr4:2:0への変換)を適用するか、又は圧縮に対してより回復性がある信号分布を得るために入力ピクチャ成分の再マッピングを行う(例え

50

ば、色成分の1つのヒストグラム平坦化を使用する)符号化前の処理(101)にかけることができる。メタデータが前処理に関連し得、ビットストリームに付加され得る。

#### 【0022】

HEVCでは、1つ又は複数のピクチャを有するビデオシーケンスを符号化するためにピクチャを1つ又は複数のスライスに分割し(102)、各スライスは、1つ又は複数のスライスセグメントを含み得る。スライスセグメントは、コード化単位、予測単位及び変換単位で構成される。HEVC仕様は、「ブロック」と「単位」とを区別し、「ブロック」は、サンプルアレイ内の特定の領域(例えば、ルマ、Y)を扱い、「単位」は、全ての符号化済み色成分(Y、Cb、Cr又はモノクロ)、構文要素及びブロックに関連する予測データ(例えば、動きベクトル)のコロケートドブロックを含む。

10

#### 【0023】

HEVCにおけるコード化のために、構成可能なサイズを有する正方形のコード化ツリーブロック(CTB)にピクチャが分割され、連続したコード化ツリーブロックの組がスライスにグループ化される。コード化ツリー単位(CTU)は、符号化済みの色成分のCTBを含む。CTBは、コード化ブロック(CB)への4分木分割の根であり、コード化ブロックは、1つ又は複数の予測ブロック(PB)に分割することができ、変換ブロック(TB)への4分木分割の根を形成する。コード化ブロック、予測ブロック及び変換ブロックに対応して、コード化単位(CU)は、予測単位(PU)及び変換単位(TU)の木構造の組を含み、PUは、全ての色成分に関する予測情報を含み、TUは、色成分ごとの残差コード化構文構造を含む。ルマ成分のCB、PB及びTBのサイズは、対応するCU、PU及びTUに当てはまる。本願では、「ブロック」という用語は、例えば、CTU、CU、PU、TU、CB、PB及びTBの何れかを指すために使用され得る。加えて、「ブロック」は、H.264/AVC又は他のビデオコード化規格内で規定されるマクロブロック及びパーティションを指すために、より広くは様々なサイズのデータのアレイを指すために使用することもできる。

20

#### 【0024】

符号器100の一例では、以下で説明するようにピクチャが符号器の要素によって符号化される。符号化されるピクチャは、ピクチャの一部を表すコード化単位で分割され(102)、処理される。各単位は、例えば、イントラモード又はインターモードを使用して符号化される。単位がイントラモードによって符号化される場合、符号器は、イントラ予測(160)を行う。インターモードでは、動き推定(175)及び動き補償(170)が行われる。符号器は、単位を符号化するためにイントラモード又はインターモードの何れを使用するかを決定し(105)、イントラ/インターの決定を例えば予測モードフラグによって示す。元の画像ブロックから予測済みブロックを減算すること(110)によって予測残差を計算する。次いで、予測残差を変換し(125)、量子化する(130)。量子化した変換係数並びに動きベクトル及び他の構文要素をエントロピーコード化して(145)、ビットストリームを出力する。符号器は、変換をスキップし、非変換残差信号に対して量子化を直接適用することができる。符号器は、変換及び量子化の両方をバイパスすることができ、即ち変換プロセス又は量子化プロセスを適用することなしに残差が直接コード化される。符号器は、符号化済みブロックを復号して、更なる予測のための参照を提供する。予測残差を復号するために量子化済みの変換係数を逆量子化し(140)、逆変換する(150)。復号済みの予測残差と予測済みブロックとを結合し(155)、画像ブロックを再構築する。例えば、デブロッキング/SAO(サンプル適応オフセット)フィルタリングを実行して符号化アーティファクトを減らすために、再構築したピクチャにインループフィルタ(165)を適用する。フィルタ済みの画像を参照ピクチャバッファ(180)に記憶する。

30

40

#### 【0025】

図2は、ビデオ復号器200の一例のブロック図を示す。この復号器200の変形形態が考えられるが、復号器200は、予期される全ての変形形態を記載することなしに、明瞭にすることを目的に以下に記載される。例えば、図1の復号器は、HEVC規格に改善

50

が加えられた H E V C 復号器又は J V E T によって開発されている J E M 復号器等の H E V C と同様の技術を使用する復号器であり得る。

#### 【 0 0 2 6 】

復号器 2 0 0 では、以下で説明するように、ビットストリームが復号器の要素によって復号される。ビデオ復号器 2 0 0 は、図 1 に記載した符号化パスと逆の復号パスを概して実行する。符号器 1 0 0 も、ビデオデータを符号化する一環としてビデオの復号を概して実行する。具体的には、復号器の入力は、ビデオ符号器 1 0 0 によって生成され得るビデオビットストリームを含み、ビデオを再構築できるようにする構文データ要素を含む。変換係数、動きベクトル及び他のコード化情報を得るためにビットストリームを最初にイントロピー復号する ( 2 3 0 )。ピクチャがどのように分割されるかをピクチャ分割情報が示す。従って、復号器は、復号したピクチャ分割情報に従ってピクチャを分割することができる ( 2 3 5 )。予測残差を復号するために変換係数を逆量子化し ( 2 4 0 )、逆変換する ( 2 5 0 )。復号した予測残差と予測済みブロックとを結合して ( 2 5 5 )、画像ブロックを再構築する。予測済みブロックは、イントラ予測 ( 2 6 0 ) 又は動き補償予測 ( 即ちインター予測 ) ( 2 7 5 ) から得ることができる ( 2 7 0 )。再構築済み画像にインテラフィルタ ( 2 6 5 ) を適用する。フィルタ済み画像を参照ピクチャバッファ ( 2 8 0 ) に記憶する。復号済みピクチャは、復号後の処理 ( 2 8 5 )、例えば逆色変換 ( 例えば、Y C b C r 4 : 2 : 0 から R G B 4 : 4 : 4 への変換 ) 又は符号化前の処理 ( 1 0 1 ) で行われた再マッピングプロセスの逆を行う逆再マッピングに更にかけることができる。復号後の処理は、符号化前の処理において導出され、ビットストリーム内でシグナリングされるメタデータを使用することができる。

#### 【 0 0 2 7 】

図 3 は、様々な態様及び実施形態が実装されるシステムの一例のブロック図を示す。システム 1 0 0 0 は、以下に記載の様々なコンポーネントを含む装置として実装することができる。本明細書に記載の態様の 1 つ又は複数を実行するように構成される。かかる装置の例は、これのみに限定されないが、パーソナルコンピュータ、ラップトップコンピュータ、スマートフォン、タブレットコンピュータ、デジタルマルチメディアセットトップボックス、デジタルテレビ受信機、パーソナルビデオ録画システム、接続された家庭用電化製品及びサーバ等の様々な電子装置を含む。システム 1 0 0 0 の要素は、単一の集積回路 ( I C )、複数の I C 及び / 又は個別コンポーネント内に単独で又は組み合わせて実装することができる。例えば、少なくとも 1 つの実施形態では、システム 1 0 0 0 の処理及び符号器 / 復号器の要素が複数の I C 及び / 又は個別コンポーネントにわたって分散される。様々な実施形態において、システム 1 0 0 0 は、例えば、通信バスを介して又は専用の入力及び / 又は出力ポートによって 1 つ又は複数の他のシステム又は他の電子装置に通信可能に結合される。様々な実施形態において、システム 1 0 0 0 は、本明細書に記載の態様の 1 つ又は複数を実装するように構成される。

#### 【 0 0 2 8 】

システム 1 0 0 0 は、例えば、本明細書に記載の様々な態様を実装するために自らの中にロードされた命令を実行するように構成される少なくとも 1 つのプロセッサ 1 0 1 0 を含む。プロセッサ 1 0 1 0 は、埋め込みメモリ、入出力インタフェース及び当技術分野で知られている他の様々な回路を含み得る。システム 1 0 0 0 は、少なくとも 1 つのメモリ 1 0 2 0 ( 例えば、揮発性メモリ装置及び / 又は不揮発性メモリ装置 ) を含む。システム 1 0 0 0 は、これのみに限定されないが、電氣的消去プログラム可能読取専用メモリ ( E E P R O M )、読取専用メモリ ( R O M )、プログラム可能読取専用メモリ ( P R O M )、ランダムアクセスメモリ ( R A M )、ダイナミックランダムアクセスメモリ ( D R A M )、スタティックランダムアクセスメモリ ( S R A M )、フラッシュ、磁気ディスクドライブ及び / 又は光ディスクドライブを含む不揮発性メモリ及び / 又は揮発性メモリを含み得る記憶装置 1 0 4 0 を含む。記憶装置 1 0 4 0 は、非限定的な例として、内蔵記憶装置、付加記憶装置 ( 着脱可能記憶装置及び着脱不能記憶装置を含む ) 及び / 又はネットワークアクセス可能記憶装置を含み得る。

10

20

30

40

50

## 【 0 0 2 9 】

システム 1 0 0 0 は、例えば、符号化済みビデオ又は復号済みビデオを提供するためにデータを処理するように構成される符号器 / 復号器モジュール 1 0 3 0 を含み、符号器 / 復号器モジュール 1 0 3 0 は、独自のプロセッサ及びメモリを含み得る。符号器 / 復号器モジュール 1 0 3 0 は、符号化及び / 又は復号機能を実行するために装置内に含まれ得るモジュールを表す。知られているように、装置は、符号化モジュール及び復号モジュールの一方又は両方を含み得る。加えて、符号器 / 復号器モジュール 1 0 3 0 は、システム 1 0 0 0 の別個の要素として実装することができるか、又は当業者に知られているようにハードウェアとソフトウェアとの組み合わせとしてプロセッサ 1 0 1 0 内に組み込まれ得る。

## 【 0 0 3 0 】

本明細書に記載の様々な態様を実行するために、プロセッサ 1 0 1 0 又は符号器 / 復号器 1 0 3 0 上にロードされるプログラムコードは、記憶装置 1 0 4 0 内に記憶され、その後、プロセッサ 1 0 1 0 によって実行するためにメモリ 1 0 2 0 上にロードされ得る。様々な実施形態によれば、プロセッサ 1 0 1 0、メモリ 1 0 2 0、記憶装置 1 0 4 0 及び符号器 / 復号器モジュール 1 0 3 0 の 1 つ又は複数は、本明細書に記載のプロセスの実行中に様々なアイテムの 1 つ又は複数を記憶し得る。記憶されるかかるアイテムは、これのみに限定されないが、入力ビデオ、復号済みビデオ又は復号済みビデオの一部、ビットストリーム、行列、変数並びに式、公式、演算及び演算ロジックの処理の中間結果又は最終結果を含み得る。

## 【 0 0 3 1 】

幾つかの実施形態では、プロセッサ 1 0 1 0 及び / 又は符号器 / 復号器モジュール 1 0 3 0 の内部のメモリを使用して命令を記憶し、符号化又は復号中に必要な処理用のワーキングメモリを提供する。しかし、他の実施形態では、これらの機能の 1 つ又は複数のために、処理装置（例えば、処理装置は、プロセッサ 1 0 1 0 又は符号器 / 復号器モジュール 1 0 3 0 であり得る）の外部のメモリが使用される。外部メモリは、メモリ 1 0 2 0 及び / 又は記憶装置 1 0 4 0、例えばダイナミック揮発性メモリ及び / 又は不揮発性フラッシュメモリであり得る。幾つかの実施形態では、例えば、テレビのオペレーティングシステムを記憶するために外部の不揮発性フラッシュメモリが使用される。少なくとも 1 つの実施形態では、MPEG-2（MPEGは、Moving Picture Experts Groupを指し、MPEG-2は、ISO/IEC 13818とも呼ばれ、13818-1は、H.222としても知られ、13818-2は、H.262としても知られる）、HEVC又はVVC等のビデオのコード化及び復号操作のワーキングメモリとしてRAM等の高速な外部のダイナミック揮発性メモリが使用される。

## 【 0 0 3 2 】

システム 1 0 0 0 の要素への入力は、ブロック 1 1 3 0 内に示す様々な入力装置によって提供され得る。かかる入力装置は、これのみに限定されないが、( i ) 例えば、ブロードキャストによって無線で伝送されるRF（無線周波数）信号を受信するRF部分、( i i ) コンポーネント（COMP）入力端子（又はCOMP入力端子の組）、( i i i ) ユニバーサルシリアルバス（USB）入力端子、及び / 又は( i v ) 高精細度マルチメディアインタフェース（HDMI）入力端子を含む。図 3 に不図示の他の例は、複合ビデオを含む。

## 【 0 0 3 3 】

様々な実施形態において、ブロック 1 1 3 0 の入力装置は、当技術分野で知られている関連する個々の入力処理要素を有する。例えば、RF部分は、( i ) 所望の周波数を選択し（信号を選択する又は信号をある周波数帯域に帯域制限するとも言う）、( i i ) 選択した信号をダウンコンバートし、( i i i ) （例えば）特定の実施形態においてチャンネルと呼ばれ得る信号周波数帯域を選択するために、より狭い周波数帯域に再び帯域制限し、( i v ) ダウンコンバート及び帯域制限済みの信号を復調し、( v ) 誤り訂正を行い、及び( v i ) データパケットの所望のストリームを選択するために逆多重化するのに適した要素に関連し得る。様々な実施形態のRF部分は、これらの機能を実行するための 1 つ又

10

20

30

40

50

は複数の要素、例えば周波数セレクタ、信号セレクタ、帯域制限器、チャンネルセレクタ、フィルタ、ダウンコンバータ、復調器、誤り訂正器及びデマルチプレクサを含む。RF部分は、例えば、受信した信号をより低い周波数（例えば、中間周波数又は基底帯域に近い周波数）又は基底帯域にダウンコンバートすることを含む、これらの機能の様々なものを行うチューナを含むことができる。あるセットトップボックスの実施形態では、RF部分及びその関連する入力処理要素が有線（例えば、ケーブル）媒体上で伝送されるRF信号を受信し、所望の周波数帯域にフィルタリングし、ダウンコンバートし、再びフィルタリングすることによって周波数の選択を行う。様々な実施形態は、上記で説明した（及び他の）要素の順序を並べ替え、それらの要素の一部を除去し、及び/又は同様の若しくは異なる機能を実行する他の要素を追加する。要素を追加することは、既存の要素間に要素を挿入すること、例えば増幅器及びアナログ-デジタル変換器を挿入することを含み得る。様々な実施形態において、RF部分は、アンテナを含む。加えて、USB及び/又はHDMI端子は、USB及び/又はHDMI接続の両端間でシステム1000を他の電子装置に接続するための個々のインタフェースプロセッサを含み得る。例えば、別個の入力処理IC内又はプロセッサ1010内において、入力処理、例えばリードソロモン誤り訂正の様々な側面を必要に応じて実装できることを理解すべきである。同様に、USB又はHDMIインタフェース処理の側面を別個のインタフェースIC内又はプロセッサ1010内で実装することができる。出力装置上で提示するためにデータストリームを必要に応じて処理するために、復調済みの、誤り訂正済みの及び逆多重化済みのストリームは、例えば、メモリ及び記憶要素と組み合わせて動作するプロセッサ1010及び符号器/復号器1030を含む様々な処理要素に与えられる。

10

20

#### 【0034】

システム1000の様々な要素を一体型ハウジング内に設けることができる。一体型ハウジング内では、様々な要素が相互接続され、適切な接続構成、例えばInter-IC（IC）バス、配線及びプリント回路基板を含む当技術分野で知られている内部バスを使用して、それらの間でデータを伝送し得る。

#### 【0035】

システム1000は、通信チャンネル1060を介して他の装置と通信することを可能にする通信インタフェース1050を含む。通信インタフェース1050は、これのみに限定されないが、通信チャンネル1060上でデータを送受信するように構成されるトランシーバを含み得る。通信インタフェース1050は、これのみに限定されないが、モデム又はネットワークカードを含むことができ、通信チャンネル1060は、例えば、有線媒体及び/又は無線媒体内に実装することができる。

30

#### 【0036】

様々な実施形態において、Wi-Fiネットワーク、例えばIEEE802.11（IEEEは、Institute of Electrical and Electronics Engineersを指す）等の無線ネットワークを使用してデータがシステム1000にストリームされるか又は与えられる。これらの実施形態のWi-Fi信号は、Wi-Fi通信に適合される通信チャンネル1060及び通信インタフェース1050上で受信される。これらの実施形態の通信チャンネル1060は、ストリーミングアプリケーション及び他のオーバーザトップ通信を可能にするためにインターネットを含む外部ネットワークへのアクセスを提供するアクセスポイント又はルータに典型的に接続される。他の実施形態は、入力ブロック1130のHDMI接続上でデータを届けるセットトップボックスを使用してストリームデータをシステム1000に与える。更に他の実施形態は、入力ブロック1130のRF接続を使用してストリームデータをシステム1000に与える。上記で示したように、様々な実施形態がデータを非ストリーミング式に提供する。加えて、様々な実施形態は、Wi-Fi以外の無線ネットワーク、例えばセルラネットワーク又はBluetoothネットワークを使用する。

40

#### 【0037】

システム1000は、ディスプレイ1100、スピーカ1110及び他の周辺装置1120を含む様々な出力装置に出力信号を与えることができる。様々な実施形態のディスプ

50

レイ 1100 は、例えば、タッチスクリーンディスプレイ、有機発光ダイオード (OLED) ディスプレイ、湾曲ディスプレイ及び/又は折り畳み式ディスプレイの 1 つ又は複数を含む。ディスプレイ 1100 は、例えば、テレビ、ラップトップ、セルホン (携帯電話) 又は他の装置であり得る。ディスプレイ 1100 は、(例えば、スマートフォンにあるように) 他のコンポーネントに統合され得るか、又は別個 (例えば、ラップトップ用の外部モニター) であり得る。実施形態の様々な例において、他の周辺装置 1120 は、独立型デジタルビデオディスク (又はデジタル多用途ディスク) (両方の用語で DVR)、ディスクプレーヤ、ステレオシステム及び/又は照明システムの 1 つ又は複数を含む。様々な実施形態は、システム 1000 の出力に基づく機能を提供する 1 つ又は複数の周辺装置 1120 を使用する。例えば、ディスクプレーヤは、システム 1000 の出力を再生する機能を実行する。

10

#### 【0038】

様々な実施形態において、AV.Link、Consumer Electronics Control (CEC) 又はユーザの介入ありの若しくはなしの装置間制御を可能にする他の通信プロトコル等のシグナリングを使用し、システム 1000 とディスプレイ 1100、スピーカ 1110 又は他の周辺装置 1120 との間で制御信号が通信される。出力装置は、個々のインタフェース 1070、1080 及び 1090 による専用接続を介してシステム 1000 に通信可能に結合され得る。代わりに、出力装置は、通信インタフェース 1050 を介して通信チャンネル 1060 を使用してシステム 1000 に接続され得る。ディスプレイ 1100 及びスピーカ 1110 は、例えば、テレビ等の電子装置内でシステム 1000 の他のコンポーネントと共に単一のユニットに一体化することができる。様々な実施形態において、ディスプレイインタフェース 1070 は、例えば、タイミングコントローラ (T Con) チップ等のディスプレイドライバを含む。

20

#### 【0039】

例えば、入力 1130 の RF 部分が別個のセットトップボックスの一部である場合、ディスプレイ 1100 及びスピーカ 1110 は、他のコンポーネントの 1 つ又は複数から代わりに切り離すことができる。ディスプレイ 1100 及びスピーカ 1110 が外部コンポーネントである様々な実施形態において、出力信号は、例えば、HDMI ポート、USB ポート又は COMP 出力を含む専用出力接続によって与えることができる。

#### 【0040】

実施形態は、プロセッサ 1010 によって実装されるコンピュータソフトウェア若しくはハードウェア又はハードウェアとソフトウェアとの組み合わせによって実行することができる。非限定的な例として、実施形態は、1 つ又は複数の集積回路によって実装され得る。メモリ 1020 は、技術的環境に適した任意の種類のものであり得、非限定的な例として、光メモリ装置、磁気メモリ装置、半導体ベースのメモリ装置、固定メモリ及び着脱可能メモリ等、任意の適切なデータ記憶技術を使用して実装することができる。プロセッサ 1010 は、技術的環境に適した任意の種類のものであり得、非限定的な例として、マイクロプロセッサ、汎用コンピュータ、専用コンピュータ及びマルチコアアーキテクチャに基づくプロセッサの 1 つ又は複数を含む。

30

#### 【0041】

図 4 は、圧縮領域内のコード化ツリー単位及びコード化ツリーの一例を示す。HEVC ビデオ圧縮規格では、ピクチャがいわゆるコード化ツリー単位 (CTU) に分割され、CTU のサイズは、例えば、 $64 \times 64$ 、 $128 \times 128$  又は  $256 \times 256$  ピクセルである。各 CTU は、圧縮領域内のコード化ツリーによって表される。これは、CTU の 4 分木分割であり、それぞれの葉をコード化単位 (CU) と呼ぶ。

40

#### 【0042】

図 5 は、CTU をコード化単位、予測単位及び変換単位に分割する一例を示す。各 CU に何らかのイントラ予測パラメータ又はインター予測パラメータ予測情報) が与えられる。そのために、1 つ又は複数の予測単位 (PU) に CU を空間的に分割し、各 PU に何らかの予測情報が指定される。イントラコード化モード又はインターコード化モードは、C

50

Uレベルで指定される。

【0043】

ビデオコーデックは、コード化単位(CU)又は変換単位と従来から名付けられている正方形ブロック又は矩形ブロックにピクチャを分割する。分割は、再帰的であり得る。従って、現在のブロックの分割深度は、現在のブロックを得るための再帰的分割の回数を指す。次いで、これらのブロックを予測することができ、残差(元と予測との差)を変換して周波数係数を計算する。符号器側では、量子化は、結果として生じる変換済みの係数を、量子化ステップサイズ(Qstep)とも名付けられる特定の量子化スケールで割ることを通常含む。次いで、復号器は、その結果に同じQstepを乗じて元の係数の近似を復元する。

【0044】

通常、量子化ステップサイズQstepを決定するために量子化パラメータ(QP)を使用する。HEVCの例では、QPは、0~51(QP範囲)の52個の値を取り得る。QPが1増加することは、量子化ステップサイズQstepが約12%(即ち $2^{1/6}$ )増加することを意味する。HEVCの例では、結果として生じるQPと、均等な量子化ステップサイズとの間の関係が以下の式によって与えられる(「イコール」記号は、整数の等価性を意味する)。

$$Qstep(QP) = (2^{1/6})^{QP-4} \quad (\text{式1})$$

【0045】

以下に記載する実施形態では、上記の(式1)で定義した以外の関数及びQP範囲を使用することができる。例えば、VVCでは、QP範囲が63に拡張されている。

【0046】

以下では、「ブロック」という用語を使用する。この用語は、例えば、ルマブロック及び/又はクロマブロック又は3つのRGBブロック等の幾つかのブロックをグループ化する「CU」という用語によって均等に置換することができる。

【0047】

現在のピクチャ(又はスライス等のその一部分)のQPからのオフセット(QPデルタ)を指定することにより、量子化ステップの局所的調節が典型的に可能である。HEVCの例では、QPデルタは、所与の最大分割深度(ピクチャパラメータセット内で認められるdiff\_cu\_qp\_delta\_depth構文要素)によって定められる「量子化グループ」(QG)と呼ばれるブロックのグループごとに指定することができ、即ち、更なる分割によって生じる全てのブロックは、同じ量子化グループに属し、同じQPを共有する(より正確には同じQP予測を共有し、最大で1つのQPデルタを指定する)。HEVCの例では、所与の分割深度における全てのブロックが同じサイズを有し、それは、全ての分割がクワッド分割(Q分割)であり、従って同じサイズのブロックについて均一なQP値をもたらすからである。

【0048】

図6は、例えば、VVCにあるような様々な分割の種類を示す。VVC等のコーデックの例では、Q分割に加えて、バイナリ分割(B分割)、ターナリ分割(T分割)及び非対称バイナリ分割(AB分割)を含む他の分割の種類があり得る。この図面では、Q分割を破線で示し、他の分割の種類を鎖線で示す。最上位のQ分割は、同じサイズの4つのブロックを画定する。第2のレベルの分割では、Q分割によって生じるブロックのそれぞれが別の種類の分割を用いて分割されており、即ち左上のブロックがAB分割を用いて分割され、右上のブロックがT分割を用いて分割され、左下のブロックがQ分割を用いて分割され、右下のブロックがB分割を用いて分割されている。これらの分割は、同じ分割深度(第2のレベル)において異なるサイズを有し得るブロックをもたらす。実際、AB分割ブロックのブロック61は、T分割のブロック62、又はQ分割のブロック63、又はB分割ブロックのブロック64よりも大きい。これらの全てのブロックは、同じレベルにあるが、これらのブロックのサイズは、同一ではなく、複数回の分割後に非常に異なり得る。従って、ブロックのサイズに少なくとも部分的に関係するブロックの視覚的重要度は、一様ではない。

10

20

30

40

50

## 【 0 0 4 9 】

図 7 A 及び図 7 B は、ブロックサイズに対する繰り返し分割の影響を示す。図 7 B は、繰り返しの水平及び垂直ターナリ分割を示す。この事例では、同じ分割深度を有しながら、中央のブロック ( 7 4 ) が角のブロック ( 7 3 ) よりもはるかに大きい。従って、量子化グループを定めるために分割深度情報を使用することは、不均一なサイズを有する量子化グループをもたらす可能性があり、ピクチャ又はスライスの全体にわたって Q P の粒度 ( 量子化グループのサイズ ) が同じである H E V C の原理と比較して不所望の結果を招き得る。図 7 A は、繰り返しの水平及び垂直 A B 分割を示し、ブロック 7 1 及びブロック 7 2 で同じ問題を引き起こす。

## 【 0 0 5 0 】

図 8 は、異なる分割レベルにおける同じサイズの様々なブロックを示す。実際、図 7 に示すように分割の種類に従って同じブロックサイズを異なる分割深度において得ることができ、図中、ブロック内の数字は、ブロックごとの分割深度を示す。これらの例では、親ブロックの 1 / 1 6 の面のブロックが 2 ( 左下のブロック 8 1 ) 又は 3 ( 右下のブロック 8 2 ) の分割深度を有する。

## 【 0 0 5 1 】

図 9 は、H E V C 仕様における量子化グループの使用を示す。H E V C 仕様は、Q P 予測とは別個の Q P デルタパーズングを含む。Q G 開始を検出することにより、新たな予測及び新たな Q P デルタが使用可能にされる。そのために、分割深度がストリーム内でシグナリングされる所与の閾値である depthLimit 以下である場合、フラグ ( IsCuQpDeltaCoded ) を偽に設定する。次いで、変換単位 ( T U ) において、ブロック残差が非空であり ( 且つ Q P オフセットのシグナリングが高レベルで有効にされており )、IsCuQpDeltaCoded が引き続き偽の場合、このフラグを真に設定し、Q P オフセットをコード化 / 復号する。復号プロセスでは、各ブロックの Q P の導出は、量子化グループの左上角の上部の及び左側の隣接ブロックを用いて行われる Q P 予測を含む。この左上角の位置は、図 9 に示すように、Q G グリッド上での現在のブロック位置の単純な位置合わせを使用して決定される。全ての分割がクワッド分割であり、従って Q G が同じ ( 最小 ) サイズを有し分割深度制限と一致するグリッド上で位置合わせされるため、この手法は、機能し、前のポイントと整合する。しかし、例えば、V V C で使用されるような上記で紹介した Q 分割と異なる分割の種類によるビデオのコード化を使用する場合、異なる分割の種類が様々なサイズのパーティションをもたらすため、Q G の位置のそのような単純な導出は、もはや正しくない。所与の分割深度におけるブロックは、もはや正規グリッドを形成しない。

## 【 0 0 5 2 】

以下に記載する実施形態は、上記の内容を念頭に置いて考案されている。

## 【 0 0 5 3 】

## 再分割レベルカウンタ

図 1 0 は、再分割レベルカウンタの使用に関する少なくとも 1 つの実施形態を示す。かかるカウンタ ( 以下では「subdiv」と呼ぶ ) は、ブロックの面に密に関係するか、又はブロックのサイズを少なくとも表し、従ってブロックの面を表すか又はブロックのサンプルの数も表す。様々なサイズの複数のブロックに画像が分割される場合、再分割レベルカウンタ ( 以下では「subdiv」と呼ぶ ) がパーティション及びブロックのそれぞれに関連付けられる。ブロックに関する subdiv 値は、少なくとも分割の種類に従って決定される。subdiv 値は、ブロックの量子化パラメータ及びブロックが属する量子化グループを決定するために使用され得る。その結果、同様のサイズのブロックに同じ量子化パラメータが使用され、従って一貫した符号化 / 復号を提供する。

## 【 0 0 5 4 】

少なくとも 1 つの実施形態では、パーティションが分割されるとき、新たな量子化グループの開始を決定するために、再分割レベルカウンタ subdiv、再分割レベルの最大値及び分割の種類に従って分析が行われ、該当する場合、パーティションの現在位置は、更に分割されるパーティションと共に記憶されるようにそれらのパーティションに伝搬され、且

10

20

30

40

50

つ復号時の予測プロセスにおいて役立つ。

【 0 0 5 5 】

少なくとも1つの実施形態では、再帰的プロセスが各ブロックに再分割レベルカウンタ  $s_{ubdiv}$  (正の整数値) を割り当てる。そのために、このプロセスは、分割木の根において  $s_{ubdiv}$  をまずゼロに初期化する。その後、分割ごとに、前の  $s_{ubdiv}$  値からインクリメントした後に作成済みの各パーティション (慣例的にサブブロックと呼ばれる) に  $s_{ubdiv}$  値を伝搬し、インクリメント値は、分割の種類及び分割内のパーティションの位置によって決まる。ブロック又はサブブロックの分割を行う間、分割深度に加えて再分割レベルが各下位パーティションに付加される。再分割レベルは、分割の種類及びパーティション数に応じて親ノードの1からインクリメントされる。

10

【 0 0 5 6 】

かかる再帰的分割機能を図10のS1によって示す。従来の分割機能に関する他の機能をS11で実行した後、分割の種類に基づいてS12で  $s_{ubdiv}$  値を決定し、各下位パーティションに  $s_{ubdiv}$  値を付加しながらS13でパーティションを再帰的に分割する。以下では、この全体的なプロセスを  $s_{ubdiv}$  プロセスと呼ぶ。

【 0 0 5 7 】

図1の符号器100、図2の復号器200及び図3のシステム1000は、本明細書に記載の実施形態の少なくとも1つを実装するように適応され、従って  $s_{ubdiv}$  プロセスによって  $s_{ubdiv}$  カウンタを扱うように構成される。

【 0 0 5 8 】

図11A、図11B及び図11Cは、一実施形態による分割の種類と  $s_{ubdiv}$  のインクリメントとの間の関係の例を示す。少なくとも1つの実施形態では、検討する分割の種類がQ分割、B分割及びT分割であり、対応するインクリメントを次の通り行う。

- Q分割では：クワッド分割の全てのパーティションについて  $incQ=+2$  (図11A)
- B分割では：水平又は垂直であり得るバイナリ分割の全てのパーティションについて  $incB=+1$  (図11B)
- T分割では：水平又は垂直であり得るターナリ分割の横のパーティションについて  $incTS=+2$ 、中央のパーティションについて  $incTC=+1$  (図11C)

20

【 0 0 5 9 】

図12は、一実施形態による  $s_{ubdiv}$  プロセスのブロック図の一例を示す。この図では、  $s_{ubdiv}$  パラメータのインクリメントをQ分割の場合に  $incQ$ 、B分割の場合に  $incB$ 、T分割の場合に  $incTS$  及び  $incTC$  で概して示し、それらの値は、上記で紹介したように定められる。ステップ401で分割の種類がQ分割であると判定された場合、ステップ403で4つ全ての下位パーティションの  $s_{ubdiv}$  値を  $incQ=+2$  インクリメントする。ステップ402で分割の種類がB分割であると判定された場合、ステップ405で両方の下位パーティションの  $s_{ubdiv}$  値を  $incB=+1$  インクリメントする。ステップ404で分割の種類がT分割であると判定された場合、ステップ407で横のパーティションの  $s_{ubdiv}$  値を  $incTS=+2$  インクリメントし、横のパーティションの  $s_{ubdiv}$  値を  $incTC=+1$  インクリメントする。

30

【 0 0 6 0 】

図13A、図13B及び図13Cは、図6及び図7の例にプロセスを適用したときの  $s_{ubdiv}$  カウンタの値の例を示す。但し、この図では、左上ブロックのAB分割を分割なしで置換しており、その理由は、図11の実施形態がAB分割のパーティションを扱わないからである。  $s_{ubdiv}$  値は、分割深度パラメータよりも実際のブロックサイズにはるかに関連していることが認められる。

40

【 0 0 6 1 】

この原理の1つの応用は、量子化中に使用され、同じ  $s_{ubdiv}$  値を有するブロックは、同じ面を有するため、それらのブロックに同数のサンプルを割り当てる。従って、少なくとも1つの実施形態では、量子化グループの仕様が検討中のコード化グループの  $s_{ubdiv}$  値に関係する。一実施形態では、量子化グループの決定は、  $s_{ubdiv}$  値をより高いレベル (例えば、PPS、スライスヘッダ、タイルグループヘッダ、タイルヘッダ内) でシグナリング

50

される最大subdiv値（本明細書ではdiff\_cu\_qp\_delta\_subdivと呼ぶ）と比較することに基づき得る。例えば、量子化グループは、subdiv = diff\_cu\_qp\_delta\_subdivの状態  
で親ブロックを分割することによって生じるブロックの組として指定される。かかるブ  
ロックの組は、同じ量子化グループを作る。上記のsubdiv導出プロセスにより、量子化グル  
ープは、同数のサンプル（=同じ面）を有するか、又はsubdivの制限に達していない（分  
割が不十分である）場合に更に大きい。例えば、図13Aでは、左下ブロック及び右上ブ  
ロックが同じ形状を有さないが、同様の面を有し、同じsubdiv値を共有する。更に、これ  
は、HEVCと同様の挙動を与える。実際、Q分割しかない場合、この方法は、HEVC  
と同じ量子化グループをもたらす。

#### 【0062】

少なくとも1つの実施形態では、ブロックのサンプル数（=面）が以下のようにsubdiv  
値から容易に計算される。

block\_surface=CTB\_surface subdiv

#### 【0063】

別の実施形態では、subdivが他の目的にも有用であり得る。一例は、subdiv値に基づ  
いて（例えば、分割モードに関する）幾つかのCABACコンテキストidを導出するこ  
とである。

#### 【0064】

##### AB分割の第1の変形形態

少なくとも1つの実施形態は、図6に示した全ての分割の種類を使用する、従って先の  
Q分割、B分割及びT分割に加えてAB分割も使用するビデオのコード化に関する。

#### 【0065】

図14A及び図14Bは、非対称バイナリ分割の異なる種類を示す。AB分割では、両  
側に関してサンプル数が2のべき乗で分割されず、即ち、比率は、3/4と1/4又は3  
/4のパーティションを同じ方向に更にAB分割する場合、2/3と1/3である。

#### 【0066】

更に、ターナリ分割に相当する連続した2回の非対称バイナリ分割は、図14Cに示す  
均等なターナリ分割と同じ再分割レベルを与えるものとする。

#### 【0067】

少なくとも1つの実施形態では、Q分割、B分割及びT分割に関して先に記載したイン  
クリメントに適合し、CTB/ブロックの面の比率を $2^{\text{subdiv}}$ で引き続き概算する方法  
は、AB分割の場合に以下のsubdivのインクリメントを適用する。

- 1/4のAB分割では：大きい部分についてincAB1=+0、小さい部分についてincA  
B2=+2

- 1/3のAB分割では：大きい部分についてincAB3=+1、小さい部分についてincA  
B4=+2

#### 【0068】

図15A及び図15Bは、AB分割に関する第1の変形実施形態による、分割の種類とs  
ubdivのインクリメントとの間の関係の例を示す。図16A、図16B及び図16Cは、  
ターナリ分割に相当する連続した2回の非対称バイナリ分割にプロセスを適用したときのs  
ubdivカウンタの値の例を示す。

#### 【0069】

##### AB分割の第2の変形形態

図17は、AB分割に関する第1の変形形態を用いた、相対する方向の連続した2回の  
非対称1/4バイナリ分割にプロセスを適用したときのsubdivカウンタの値の一例を示す  
。この事例では、図17に示すように、相対する方向の連続した1/4のAB分割は、サイ  
ズの著しい減少（元の面の56%）にもかかわらず、大きい部分のsubdivがインクリメ  
ントされないことを招く（右上ブロック）。

#### 【0070】

図18A、図18B、図18C、図18D及び図18Eは、AB分割に関する第2の変

10

20

30

40

50

形実施形態による、分割の種類とsubdivのインクリメントとの間の関係の例を示す。この実施形態では、以下のように全ての分割の種類についてsubdivのインクリメントが修正される。

- クワッド分割の全てのパーティションについてincQ=+4 (図18A)、
- 水平バイナリ分割 (図18B) 又は垂直 (不図示) の全てのパーティションについてincB=+2、
- ターナリ分割 (図18Cに示すように垂直又は水平であり得る) の横のパーティションについてincTS=+4、中央のパーティションについてintT2=+2、
- 垂直の1/4のAB分割に関して図18Dに示すように、1/4のAB分割の大きい部分についてincAB1=+1、小さい部分についてincAB2=+4 (水平も同様になる)、
- 垂直の1/3のAB分割に関して図18Eに示すように、1/3のAB分割の大きい部分についてincAB3=+1、小さい部分についてincAB4=+3 (水平も同様になる)。

#### 【0071】

この変形形態を使用し、均等なAB分割及びターナリ分割は、依然として同じsubdivを取得し、CTB/ブロックの面の比率は、 $2^{(subdiv/2)}$ に近い。従って、ブロックの面の概算を計算するために次の簡単な計算を使用することができる：`block_surface CTB_surface (subdiv/2)`。非対称バイナリ分割が存在する場合、同じsubdivを有するパーティションは、第1の変形形態よりもサイズが近い。

#### 【0072】

量子化グループの画定への影響

図9に関して既に論じたように、QPデルタパーズィングのために及びQP予測のために、量子化グループの開始 (以下ではQG開始と呼ぶ) の統一された検出が要求される。少なくとも1つの実施形態では、QG開始の検出を以下のように行う。`cu_qp_delta_enabled_flag` (デルタQPコーディングの使用を活性化又は非活性化できるようにするフラグ) 及び`cu_qp_delta_subdiv` (QGを設定するために使用されるsubdivの制限を指定するパラメータ) という2つの高レベル構文要素をビットストリーム (例えば、PPS、スライスヘッダ、タイルグループヘッダ、タイルヘッダ内) から読み出す。`cu_qp_delta_enabled_flag`が真に等しい場合に下記が該当する。

#### 【0073】

分割プロセス中、`subdiv = cu_qp_delta_subdiv`である場合、QG開始を示すためにフラグ (`CuQpDeltaCoding`) を真に設定し、現在のQG原点として現在位置を保存する。QPデルタがコード化 (又は復号) されると直ちに、`CuQpDeltaCoding`フラグを (偽に) リセットする。

#### 【0074】

しかし、新たな分割の種類を導入することにより、T分割又はAB分割によって生じる様々なサイズのパーティションが原因でQGの画定が非自明になる。`cu_qp_delta_subdiv`以下のsubdiv値を有する最小パーティションとしてQGを簡単に画定することはできない。

#### 【0075】

図19は、T分割の場合のQG開始の一例を示す。この例では、問題を簡潔に示す目的で`cu_qp_delta_subdiv`の制限値を1に設定する。図11Cで定めたsubdivのインクリメントを使用し、親のパーティションが`subdiv=0`を有したと仮定し、T分割は、横のパーティションについて`subdiv=2`、中央のパーティションについて1をもたらす (図19の左側)。中央のパーティションは、QGであるべきであり (そのsubdivは、制限内：`subdiv = cu_qp_delta_subdiv`)、横のパーティションは (そのsubdiv値が`cu_qp_delta_subdiv`を上回るため) QGグループであるべきではなく、横のパーティションが属するQGは、その親のパーティションである (又はない) ことを示唆している。これは、重複するQG (又はQG内に穴があること) を招き、それは、あり得ないことである (ブロックは、単一のQP予測を有するものとし、即ち1つのみのQGに属するべきである)。この問題は、以下で説明するように訂正可能である。

10

20

30

40

50

## 【 0 0 7 6 】

図 2 0 は、Q G の問題を訂正する一実施形態による T 分割の場合の Q G の画定の一例を示す。この実施形態によれば、T 分割では、

- 最初のパーティションの subdiv 値が `cu_qp_delta_subdiv` を上回り、そのため、Q G 開始フラグは、修正されていないが、その位置を Q G 原点とした状態で、Q G 開始フラグは、親のパーティションによって既に真に設定されている。
- 中央のパーティションについて、その subdiv 値が `cu_qp_delta_subdiv` に等しく、従って Q G 開始フラグが真に設定される。これは、新たな Q G を中央の位置において開始し、それは、最初のパーティション（左側）が完了した Q G であることを意味する。
- 最後のパーティションの subdiv 値は、`cu_qp_delta_subdiv` を上回り、従って、Q G 開始フラグは、設定されない。これは、最後のパーティションが前の Q G の要素であることを意味し、即ち中央のパーティションと最後のパーティションとが単一の Q G を形成する。

10

## 【 0 0 7 7 】

A B 分割に関して、

- 小さいパーティションが最初の場合、2 つの Q G があり、
- 小さいパーティションが最後の場合、単一の Q G があり、
- 連続した 2 回の A B 分割は、T 分割に相当する。

## 【 0 0 7 8 】

以下の節は、V V C におけるこれらの構文関数の例として JVET-L1001 の文献に記載されている構文を使用する（関数 `coding_tree_unit()` を明記する 7 . 3 . 4 . 2 節、関数 `coding_quadtree()` を明記する 7 . 3 . 4 . 3 節、関数 `multi_type_tree()` を明記する 7 . 3 . 4 . 4 節）。

20

## 【 0 0 7 9 】

図 2 1 は、Q G 識別プロセスの実施形態の一例のブロック図を示す。この実施形態では、`coding_quadtree()` 構文及び `multi_type_tree()` 構文（又は以下で `coding_tree` とまとめて呼ぶ均等な構文）に `subdiv` パラメータが追加されている。構文の変更を最小限に抑えるために、Q G 検出フラグを H E V C 風に、即ち上記で提案した `CuQpDeltaCoding` の代わりに `IsCuQpDeltaCoded` フラグを保持して示している。

## 【 0 0 8 0 】

ステップ 5 0 1 では、`subdiv` パラメータを 0 とした状態で分割を C T U から開始する。コード化ツリーレベルにおいて、ステップ 5 0 2 では、`cu_qp_delta_enabled_flag` が有効で、`subdiv = cu_qp_delta_subdiv` である場合、`IsCuQpDeltaCoded=0`、`CuQpDeltaVal=0` を含め、一部の値がステップ 5 0 3 で初期化され、ブロックの現在位置が Q G 原点、即ち  $(x_{Qg}, y_{Qg}) = (x_0, y_0)$  として記憶される。ステップ 5 0 4 は、再分割が該当するかどうかを確認する。再分割が該当する場合、ステップ 5 0 5 で分割の種類によるパラメータと共に子の分割を呼び出す（`coding_tree`）。対応する下位パーティションについて、下位パーティションごとの `subdiv` パラメータを上記の実施形態の 1 つによる下位パーティションの種類に対応する値 `incX` インクリメントする（例えば、B 分割では `incX` が + 1 であり、Q 分割では + 2 等である）。ステップ 5 0 4 でそれ以上の分割が適用されない場合（コード化ツリーの葉に到達したことを意味する）、このプロセスは、C U 及び T U のレベルに移る。ステップ 5 0 6 では、`cu_qp_delta_enabled_flag` が 1 に等しく、`IsCuQpDeltaCoded` が 0 に等しく、Q P デルタをコード化するための条件が満たされている（例えば、非ゼロ係数がコード化される = `cbf` が非ゼロである）場合、ステップ 5 0 7 で Q P デルタをコード化（又は復号）し、`CuQpDeltaVal` を適宜更新し、`IsCuQpDeltaCoded` を 1 に設定する。さもなければ、ステップ 5 0 8 では具体的なアクションを行わない。

30

40

## 【 0 0 8 1 】

図 2 2 は、Q G 識別プロセスの実施形態の別の例のブロック図を示す。この実施形態では、Q G 開始の検出に補足条件を追加することによって新たな Q G の作成を更に抑えることを提案する。子ノードが制限を上回ることを検出するとき、親ノードは、全ての子につ

50

いてQ G開始の検出を無効にする。これは、より多くの規則的Q Gを与えることをもたらす。

#### 【0082】

そのために、Q Gフラグがパラメータとしてcoding\_tree()構文に渡される。ステップ601では、0をsubdivパラメータとし、1をQ GフラグパラメータとしてCTUから分割を開始する。コード化ツリーレベルにおいて、ステップ602でQ Gフラグが真に等しく、subdiv = cu\_qp\_delta\_subdivである場合、CuQpDeltaCoding=cu\_qp\_delta\_enabled\_flag、CuQpDeltaVal=0を含め、一部の値がステップ603で初期化され、ブロックの現在位置がQ G原点、即ち(xQg,yQg)=(x0,y0)として記憶される。ステップ604は、再分割が該当するかどうかを確認する。再分割が該当する場合、ステップ605で分割の種類によるパラメータと共に子の分割を呼び出す(coding\_tree)。対応する下位パーティションについて、下位パーティションごとのsubdivパラメータを上記の実施形態の1つによる値incXであって、Xは、下位パーティションの種類である、値incX(例えば、B分割では+1)インクリメントする。次の分割が制限を超えることができず、そのため、subdiv + incMax = cu\_qp\_delta\_subdivが成立する場合にQ Gフラグパラメータを設定し、incMaxは、検討される次の分割についてあり得る最大のインクリメントである。ステップ604でそれ以上の分割が適用されない場合(コード化ツリーの葉に到達したことを意味する)、このプロセスは、CU及びTUのレベルに移る。ステップ606では、CuQpDeltaCodingが1に等しく、QPデルタをコード化するための条件が満たされている(例えば、非ゼロ係数がコード化される=cbfが非ゼロである)場合、ステップ607でQPデルタをコード化(又は復号)し、CuQpDeltaValを適宜更新し、CuQpDeltaCodingを0に設定する。さもなければ、ステップ608では具体的なアクションを行わない。

#### 【0083】

図23は、図19の状況におけるかかる実施形態の結果を示す。ターナリ分割のレベルにおいてQ G間の不整合がなくなり、従ってより優れた結果をもたらす。換言すれば、所与のノードの下位パーティションの何れかがsubdiv制限を上回るsubdiv値を有する場合、所与のノードの全ての下位パーティションが新たなQ Gを開始するのを防ぐことを提案する。これは、qgEnableと呼ばれる新たなパラメータを定めることによって行われ、multi\_type\_tree()構文関数に渡される。qgEnableは、最初、真に等しいが、横のパーティションが、subdiv制限を上回るsubdiv値を有する場合、ターナリ分割の全ての子について偽に等しく設定される。qgEnableは、新たなQ Gを開始する(及び結果としてQ G原点を保存しQPデルタコーディングを可能にする)ための追加条件として使用される。条件を併合し、構文を単純化するために、IsCuQpDeltaCodedの論理を反転させることも提案する(CuQpDeltaCodingに改称される)。

#### 【0084】

量子化グループを使用する他のコード化及び復号プロセスは、同じ利点を有する同じ原理を使用することができる。例えば、multi\_type\_tree内でcoding\_quadtreeが見つからないVVCでは、Q Gパラメータの使用をmulti\_type\_treeに制限することができる。

#### 【0085】

更に、xQg及びyQgが現在のブロック位置から導出されるのではなく、与えられることを除き、QPの導出プロセスは、HEVCと変わらないことができる(JCTVC-W1005-v4に記載のHEVC仕様の段落8.6.1)。

#### 【0086】

少なくとも1つの実施形態は、Q G開始の検出時に1に設定されるCuQpPredict内部フラグを追加することによって可能になるオンザフライのCU QP導出を使用する(例えば、CuQpDeltaCodingが1に等しい場合、xQg、yQg及びCuQpDeltaVal変数は、不要であり、削除することができる); CU内でCuQpPredictが設定される場合、QPを予測し、それをQpY変数に割り当て、CuQpPredictフラグをリセットする。TU内でCuQpDeltaCodingが1でありQPデルタが存在する場合、QpYは復号済みのデルタの値によってインクリメントされ、CuQpDeltaCodingはリセットされる。

## 【 0 0 8 7 】

## 構文要素及びセマンティクス

少なくとも1つの実施形態では、subdivプロセスがsubdivレベルカウンタの値をシグナリングする特定の構文を使用することによって実装される。以下に記載するコード化された構文要素は、HEVC又はVVC（[JVET\_K1001]を参照されたい）に基づき、変更箇所は、灰色の背景色で強調表示してある。

## 【 0 0 8 8 】

最初の変更では、diff\_cu\_qp\_delta\_depthをcu\_qp\_delta\_subdivに改称し、それに応じてセマンティクスの記述を改変した。導出変数Log2MinCuQpDeltaSizeを削除した。

## 【 0 0 8 9 】

図24は、再分割レベルカウンタを使用する実施形態による、ピクチャパラメータセット（PPS）の構文の一例を表すテーブルを示す。このテーブルでは、1に等しいcu\_qp\_delta\_enabled\_flagは、cu\_qp\_delta\_subdiv構文要素がPPS内にあり、cu\_qp\_delta\_absが変換単位の構文内にあり得ることを規定する。0に等しいcu\_qp\_delta\_enabled\_flagは、cu\_qp\_delta\_subdiv構文要素がPPS内になく、cu\_qp\_delta\_absが変換単位の構文内がないことを規定する。cu\_qp\_delta\_subdivは、cu\_qp\_delta\_abs及びcu\_qp\_delta\_sign\_flagを伝達するコード化単位の最大cbSubdiv値を規定する。存在しない場合、cu\_qp\_delta\_subdivの値は、0に等しいものと推論する。

## 【 0 0 9 0 】

## 主な実施形態のための構文要素及びセマンティクス

図25～図28は、主な実施形態を実装するための構文の例を表すテーブルを示す。構文は、VVC構文に関して図示し、修正箇所は、灰色の背景色で強調表示してある。他のパラメータと表記法を一貫させるために、subdivレベルカウンタをcbSubdivパラメータによって表す。SPLIT\_BT\_HOR、SPLIT\_BT\_VER、SPLIT\_TT\_HOR及びSPLIT\_TT\_VERは、水平B分割、垂直B分割、水平T分割及び垂直T分割にそれぞれ対応する。IsCuQpDeltaCoded及びCuQpDeltaValをリセットするためにコード化ツリー構文内で使用する条件が(cu\_qp\_delta\_enabled\_flag && log2CbSize = Log2MinCuQpDeltaSize)から(cu\_qp\_delta\_enabled\_flag && cbSubdiv = cu\_qp\_delta\_subdiv)に修正されている。この条件下において、IsCuQpDeltaCoded及びCuQpDeltaValをリセットすることに加えて、x0及びy0をxQg及びyQgに保存する（図21のステップ503及び図22のステップ603）。主な実施形態又は主な実施形態の変形形態（AB分割が必要な場合）に記載の通り、新たに作成したcbSubdivパラメータを分割の種類及びパーティションのインデックスに従ってインクリメントするために、coding\_quadtree()及びmulti\_type\_tree()（更にdual\_tree\_implicit\_qt\_split）に構文又はセマンティクスを追加する。

## 【 0 0 9 1 】

VVCの「derivation process for quantization parameters」の段落では、HEVCに記載されているようにQG原点の位置(xQg, yQg)は、Log2MinCuQpDeltaSizeに基づいていたため、導出できないが、(xQg, yQg)は、正にcoding\_quadtree()及びmulti\_type\_tree()内に保存されるものである。

## 【 0 0 9 2 】

## QG開始の検出を制限した主な実施形態のための構文要素及びセマンティクス

少なくとも1つの実施形態では、subdivプロセスは、subdivレベルカウンタの値をシグナリングする特定の構文及び量子化グループの開始を検出することに関係する少なくとも1つのパラメータを使用することによって実装される。従って、QG開始の検出は、コード化ツリー構文内でQGフラグパラメータ（以下の表では「QG」と改称する）を追加し、QG開始の検出条件を修正することによって行われる。修正箇所は、灰色の背景色で強調表示してある。

## 【 0 0 9 3 】

図29～図31は、主な実施形態を実装し、QG開始検出を処理するための構文の例を表すテーブルを示す。構文は、VVC構文に関して図示し、修正箇所は、灰色の背景色で

10

20

30

40

50

強調表示してある。coding\_quadtree()にあるのと同じ構文がdual\_tree\_implicit\_qt\_split()に使用され、ここでは図示していない。

【0094】

図32～図35は、主な実施形態を実装し、(テーブル内のイタリック体の太字で示す)クロマ関連要素に関するQG開始検出を処理するための構文の例を表すテーブルを示す。実際、上記の説明は、ルマ量子化パラメータに関連して行われたが、必要な変更を加えて同じ原理がクロマ量子化にも当てはまる。

【0095】

構文構造及び要素名は、規格のバージョンに応じて複数の変形形態で存在し得ることを当業者であれば理解するであろう。構文要素は、同じ情報を表す変数によって置換され得る。例えば、VVCの以前のバージョンは、PPS内にあるcu\_qp\_delta\_subdiv構文要素を使用したのに対し、VVCの更なるバージョンは、同じ情報を表すがピクチャヘッダ内にある構文要素に基づいて、現在のスライスタイプについて適切な値を選択する、スライスヘッダのセマンティクス内で定められるCuQpDeltaSubdiv変数を使用する。加えて、VVCの様々なバージョンにおいて、コード化ツリーの構文構造に関する幾らかの変更がある。それらの変更は、上記の実施形態の原理に影響を与えない。加えて、それらの実施形態に関連する構文は、異なる構造内の異なる位置に含まれ得る。

【0096】

様々な実装形態が復号することを含む。本願で使用する時、「復号する」は、例えば、表示に適した最終出力をもたらすために受信済みの符号化シーケンスに対して実行されるプロセスの全て又は一部を包含し得る。様々な実施形態において、かかるプロセスは、復号器によって典型的に行われるプロセス、例えばエントロピー復号、逆量子化、逆変換及び差分復号の1つ又は複数を含む。様々な実施形態において、かかるプロセスは、本願に記載の様々な実装形態の復号器によって行われるプロセス、例えば図12、図21及び図22に示したプロセスを更に又は代わりに含む。

【0097】

更なる例として、一実施形態では「復号」がエントロピー復号のみを指し、別の実施形態では「復号」が差分復号のみを指し、別の実施形態では「復号」がエントロピー復号と差分復号との組み合わせを指す。「復号プロセス」という語句が操作のサブセットを具体的に指すことを意図するか、又はより広範な復号プロセスを概して指すことを意図するかは、具体的な説明の文脈に基づいて明らかになり、当業者によって十分に理解されることが考えられる。

【0098】

様々な実装形態が符号化することを含む。「復号」に関する上記の解説と同様に、本願で使用する時、「符号化する」は、例えば、符号化ビットストリームをもたらすために入力ビデオシーケンスに対して実行されるプロセスの全て又は一部を包含し得る。様々な実施形態において、かかるプロセスは、符号器によって典型的に行われるプロセス、例えば分割、差分符号化、変換、量子化及びエントロピー符号化の1つ又は複数を含む。様々な実施形態において、かかるプロセスは、本願に記載の様々な実装形態の符号器によって行われるプロセス、例えば図12、図21及び図22のプロセスを更に又は代わりに含む。

【0099】

更なる例として、一実施形態では「符号化」がエントロピー符号化のみを指し、別の実施形態では「符号化」が差分符号化のみを指し、別の実施形態では「符号化」が差分符号化とエントロピー符号化との組み合わせを指す。「符号化プロセス」という語句が操作のサブセットを具体的に指すことを意図するか、又はより広範な符号化プロセスを概して指すことを意図するかは具体的な説明の文脈に基づいて明らかになり、当業者によって十分に理解されることが考えられる。

【0100】

本明細書で使用した構文要素は、記述用語であることに留意されたい。そのため、それらは、他の構文要素名の使用を排除しない。

10

20

30

40

50

## 【0101】

図面が流れ図として示されている場合、それは、対応する機器のブロック図も提供することを理解すべきである。同様に図面がブロック図として示されている場合、それは、対応する方法／プロセスの流れ図も提供することを理解すべきである。

## 【0102】

本明細書に記載した実装形態及び態様は、例えば、方法若しくはプロセス、機器、ソフトウェアプログラム、データストリーム又は信号によって実装することができる。単一形式の実装形態に関連してのみ論じられていても（例えば、方法としてのみ論じられていても）、論じられた特徴の実装形態は、他の形態（例えば、機器又はプログラム）でも実装することができる。機器は、例えば、適切なハードウェア、ソフトウェア及びファームウェアによって実装することができる。方法は、例えば、プロセッサによって実装ことができ、プロセッサは、例えば、コンピュータ、マイクロプロセッサ、集積回路又はプログラム可能論理装置を含む処理装置全般を指す。プロセッサは、例えば、コンピュータ、携帯電話、ポータブル／携帯情報端末（「PDA」）及びエンドユーザ間の情報の通信を促進する他の装置等の通信装置も含む。

10

## 【0103】

「一実施形態」、又は「ある実施形態」、又は「一実装形態」、又は「ある実装形態」及びそれらの他の変化形に言及することは、実施形態に関連して記載される特定の特性、構造、特性等が少なくとも1つの実施形態に含まれることを意味する。従って、本願の全体を通して様々な箇所に出現する「一実施形態では」、又は「ある実施形態では」、又は「一実装形態では」、又は「ある実装形態では」という語句及び他の任意の変化形の出現は、必ずしも全て同じ実施形態を指すものではない。

20

## 【0104】

加えて、本願は、様々な情報片を「決定すること」に言及する場合がある。情報を決定することは、例えば、情報を推定すること、情報を計算すること、情報を予測すること又は情報をメモリから取り出すことの1つ又は複数を含み得る。

## 【0105】

更に、本願は、様々な情報片に「アクセスすること」に言及する場合がある。情報にアクセスすることは、例えば、情報を受信すること、情報を（例えば、メモリから）取り出すこと、情報を記憶すること、情報を移動すること、情報を複製すること、情報を計算すること、情報を決定すること、情報を予測すること又は情報を推定することの1つ又は複数を含み得る。

30

## 【0106】

加えて、本願は、様々な情報片を「受信すること」に言及する場合がある。受信することは、「アクセスすること」と同様に広義語であることを意図する。情報を受信することは、例えば、情報にアクセスすること又は情報を（例えば、メモリから）取り出すことの1つ又は複数を含み得る。更に、「受信すること」は、典型的には、例えば情報を記憶する操作、情報を処理する操作、情報を伝送する操作、情報を移動する操作、情報を複製する操作、情報を消去する操作、情報を計算する操作、情報を決定する操作、情報を予測する操作又は情報を推定する操作等の操作中に何らかの形式で関与する。

40

## 【0107】

例えば、「A/B」、「A及び/又はB」並びに「A及びBの少なくとも1つ」の場合に「/」、「及び/又は」並びに「~の少なくとも1つ」の何れかを使用することは、最初に挙げられる（A）の選択肢のみを選択すること、又は2番目に挙げられる（B）の選択肢のみを選択すること、又は（A及びB）の両方の選択肢を選択することを包含することを意図する。更なる例として、「A、B及び/又はC」並びに「A、B及びCの少なくとも1つ」の場合、かかる表現法は、最初に挙げられる（A）の選択肢のみを選択すること、又は2番目に挙げられる（B）の選択肢のみを選択すること、又は3番目に挙げられる（C）の選択肢のみを選択すること、又は最初に挙げられる選択肢及び2番目に挙げられる選択肢（A及びB）のみを選択すること、又は最初に挙げられる選択肢及び3番目に

50

挙げられる選択肢（A及びC）のみを選択すること、又は2番目に挙げられる選択肢及び3番目に挙げられる選択肢（B及びC）のみを選択すること、又は3つ全ての選択肢（A及びB及びC）を選択することを包含することを意図する。当業者に明らかであるように、この表現法は、挙げられているアイテムの数だけ拡張することができる。

#### 【0108】

更に本明細書で使用するとき、「シグナリング」という用語は、とりわけ対応する復号器に何らかを示すことを指す。例えば、特定の実施形態では、符号器が特定のモードをシグナリングする。このようにして、一実施形態では、符号器側及び復号器側の両方において同じパラメータが使用される。従って、例えば、符号器は、特定のパラメータを復号器に伝送することができ（明確なシグナリング）、それにより復号器は同じ特定のパラメータを使用することができる。逆に、復号器が他のパラメータと共にその特定のパラメータを既に有する場合、単にその特定のパラメータを復号器が知り、選択できるようにするためにシグナリングを伝送なしに使用することができる（暗黙のシグナリング）。任意の実際の機能を伝送することを回避することにより、様々な実施形態においてビットの節約が実現される。シグナリングは、様々な方法で実現できることを理解すべきである。例えば、様々な実施形態において、対応する復号器に情報をシグナリングするために、1つ又は複数の構文要素、フラグ等が使用される。上記の内容は、「シグナル」という単語の動詞の形態に関するが、「シグナル」という単語は、本明細書では名詞としても使用され得る。

10

#### 【0109】

当業者に明白であるように、実装形態は、例えば、記憶又は伝送され得る情報を運ぶようにフォーマットされる様々な信号をもたらすことができる。情報は、例えば、方法を実行するための命令又は記載した実装形態の1つによって作り出されるデータを含み得る。例えば、信号は、記載した実施形態のビットストリームを運ぶようにフォーマットされ得る。かかる信号は、例えば、電磁波として（例えば、スペクトルの無線周波数部分を用いて）又はベースバンド信号としてフォーマットされ得る。フォーマットすることは、例えば、データストリームを符号化し、符号化データストリームで搬送波を変調することを含み得る。信号が運ぶ情報は、例えば、アナログ情報又はデジタル情報であり得る。信号は、知られているように、様々な異なる有線リンク又は無線リンク上で伝送され得る。信号は、プロセッサ可読媒体上に記憶され得る。

20

#### 【0110】

幾つかの実施形態を記載してきた。これらの実施形態の特徴は、単独で又は任意の組み合わせで提供され得る。更に、実施形態は、様々な請求項のカテゴリ及び種類にわたり、以下の特徴、装置又は態様の1つ又は複数を単独で又は任意の組み合わせで含み得る。

30

#### [付記1]

ビデオをコード化する方法であって、

- 画像のブロックがサブブロックに分割される（504）とき、
  - 前記サブブロックの面を表す値を関連付けること（505）、及び
  - 前記サブブロックの前記面を表す前記値に基づいて、前記サブブロックの量子化パラメータを決定すること
- を含む方法。

40

#### [付記2]

ビデオを復号する方法であって、

- 画像のブロックがサブブロックに分割される（504）とき、
  - 前記サブブロックの面を表す値を関連付けること（505）、及び
  - 前記サブブロックの前記面の前記面を表す前記値に基づいて、前記サブブロックの逆量子化パラメータを決定すること
- を含む方法。

#### [付記3]

ビデオを符号化する装置であって、

- 画像のブロックがサブブロックに分割される（504）とき、

50

- 前記サブブロックの面を表す値を関連付けること(505)、及び  
 - 前記サブブロックの前記面の前記面を表す前記値に基づいて、前記サブブロックの量子化パラメータを決定すること  
 を行う手段を含む装置。

[付記4]

ビデオを復号する装置であって、  
 - 画像のブロックがサブブロックに分割される(504)とき、  
 - 前記サブブロックの面を表す値を関連付けること(505)、及び  
 - 前記サブブロックの前記面の前記面を表す前記値に基づいて、前記サブブロックの逆量子化パラメータを決定すること  
 を行う手段を含む装置。

10

[付記5]

前記サブブロックの前記面を表す前記値は、画像のブロックがサブブロックに分割されるとき、前記サブブロックの分割の種類を表す値だけインクリメントされる(403、405、407)再分割レベルカウンタである、付記1若しくは2に記載の方法又は付記3若しくは4に記載の装置。

[付記6]

- サブブロックの前記再分割レベルカウンタが再分割の最大値以下である間、前記分割の種類に従って新たな量子化グループを有効にすること  
 を更に含む、付記5に記載の方法又は装置。

20

[付記7]

- 新たな量子化グループが有効にされるとき、前記ブロックの位置を記憶すること(503)  
 を更に含む、付記6に記載の方法又は装置。

[付記8]

- 画像のブロックがターナリ分割に従ってサブブロックに分割され、且つ最も小さいサブブロックの前記再分割レベルカウンタが再分割の最大値を上回るとき、新たな量子化グループを無効にすること  
 を更に含む、付記6又は7に記載の方法又は装置。

[付記9]

前記インクリメント値は、  
 クワッド分割の全てのサブブロックについて、2、  
 バイナリ分割の全てのサブブロックについて、1、  
 ターナリ分割の横のサブブロックについて、2、及び  
 ターナリ分割の中央のサブブロックについて、1  
 に等しい、付記5～8の何れか一項に記載の方法又は付記5～8の何れか一項に記載の装置。

30

[付記10]

前記インクリメント値は、  
 クワッド分割の全てのサブブロックについて、4、  
 バイナリ分割の全てのサブブロックについて、2、  
 ターナリ分割の横のサブブロックについて、4、及び中央のサブブロックについて、2、  
 非対称1/4バイナリ分割の最小のサブブロックについて、4、及び最大のサブブロックについて、1、並びに  
 非対称1/3バイナリ分割の最小のサブブロックについて、3、及び最大のサブブロックについて、1  
 に等しい、付記5～8の何れか一項に記載の方法又は付記5～8の何れか一項に記載の装置。

40

[付記11]

記憶媒体であって、サブブロックの面を表す値を表す少なくとも構文データ要素を含む

50

、その上で符号化されたビデオビットストリームデータを有し、前記サブブロックは、ビデオの画像のブロックを分割することの結果である、記憶媒体。

[付記 1 2]

前記サブブロックの前記面を表す前記値は、画像のブロックがサブブロックに分割されるとき、前記サブブロックの分割の種類を表す値だけインクリメントされる再分割レベルカウンタである、付記 1 1 に記載の記憶媒体。

[付記 1 3]

前記インクリメント値は、  
クワッド分割の全てのサブブロックについて、2、  
バイナリ分割の全てのサブブロックについて、1、  
ターナリ分割の横のサブブロックについて、2、及び  
ターナリ分割の中央のサブブロックについて、1  
に等しい、付記 1 2 に記載の記憶媒体。

[付記 1 4]

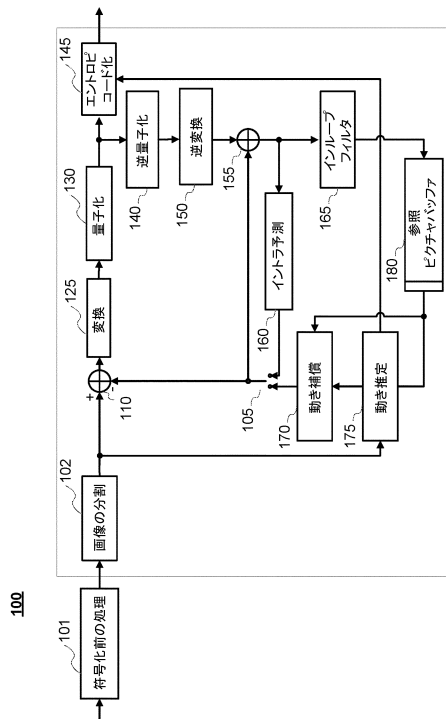
付記 1、2 又は 5 ~ 1 0 の何れか一項に記載の方法のステップを実施する、プロセッサ ( 1 0 1 0 ) によって実行可能なプログラムコード命令を含むコンピュータプログラム。

[付記 1 5]

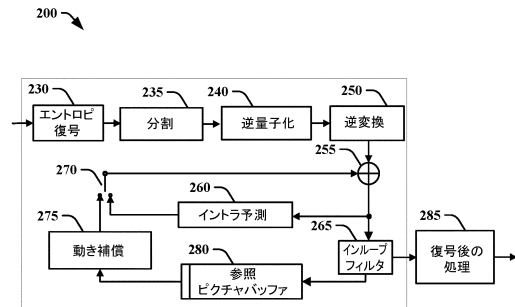
非一時的コンピュータ可読媒体上に記憶され、且つプロセッサ ( 1 0 1 0 ) によって実行されるとき、付記 1、2 又は 5 ~ 1 0 の何れか一項に記載の方法を装置 ( 1 0 0 0 ) が実施することを可能にするプログラムコード命令を含むコンピュータプログラム製品。

【図面】

【図 1】



【図 2】



10

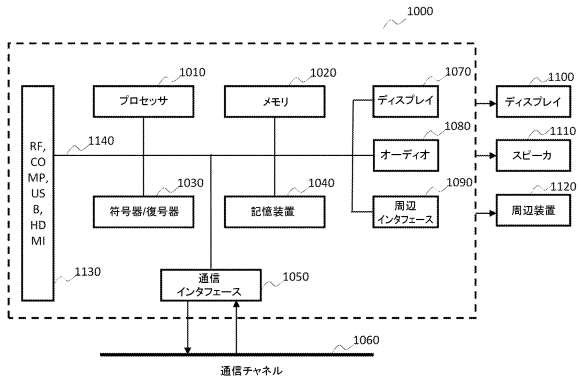
20

30

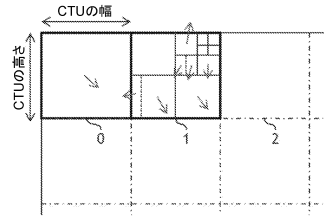
40

50

【図3】

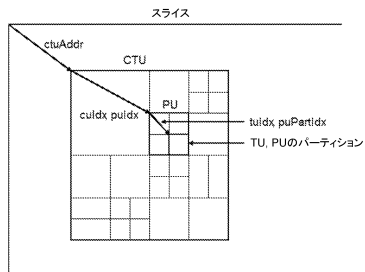


【図4】

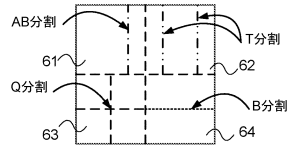


10

【図5】



【図6】



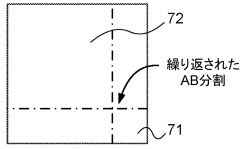
20

30

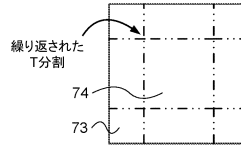
40

50

【図 7 A】

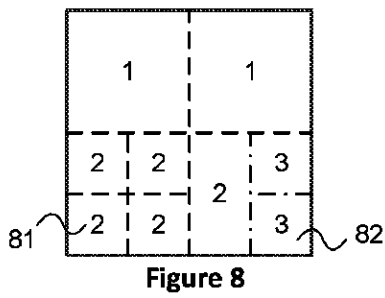


【図 7 B】

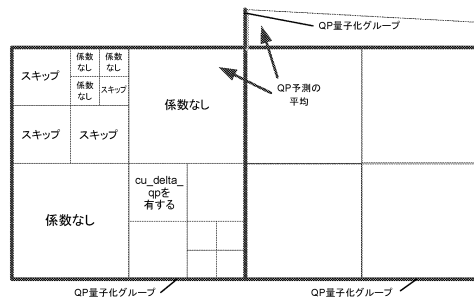


10

【図 8】

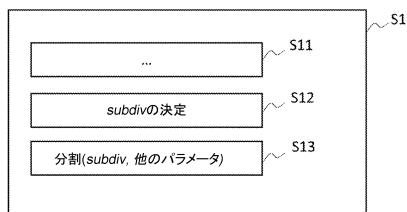


【図 9】

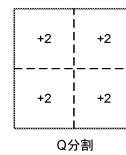


20

【図 1 0】



【図 1 1 A】



30

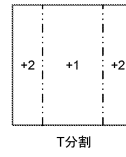
40

50

【 図 1 1 B 】

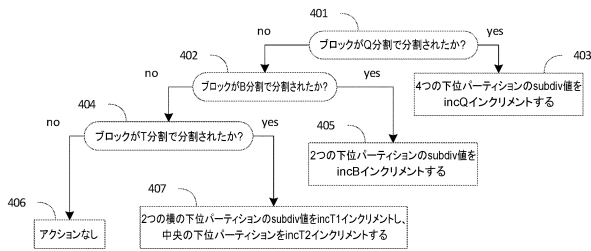


【 図 1 1 C 】



10

【 図 1 2 】



【 図 1 3 A 】

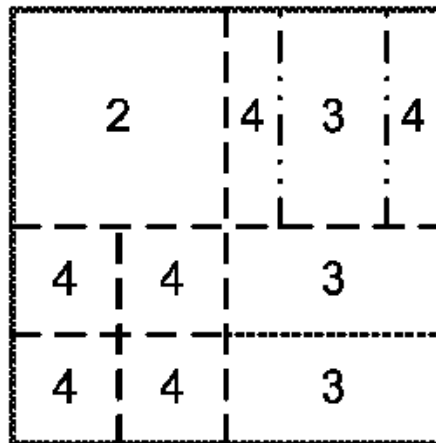


Figure 13A

20

30

40

50

【図 1 3 B】

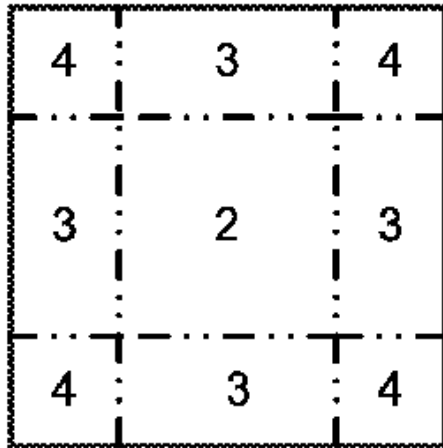


Figure 13B

【図 1 3 C】

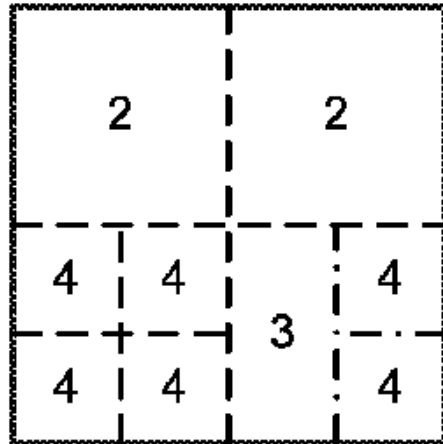
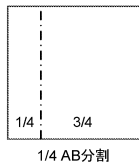


Figure 13C

10

20

【図 1 4 A】



【図 1 4 B】

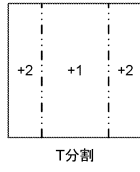


30

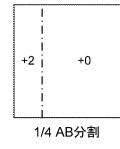
40

50

【図 1 4 C】

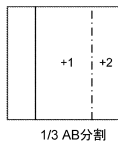


【図 1 5 A】



10

【図 1 5 B】

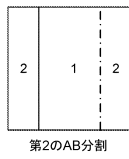


【図 1 6 A】

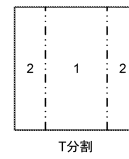


20

【図 1 6 B】



【図 1 6 C】

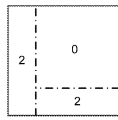


30

40

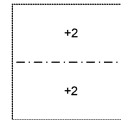
50

【図 17】



垂直の  
1/4 AB分割

【図 18 B】



B分割

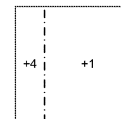
10

【図 18 C】



T分割

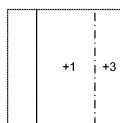
【図 19 D】



1/4 AB分割

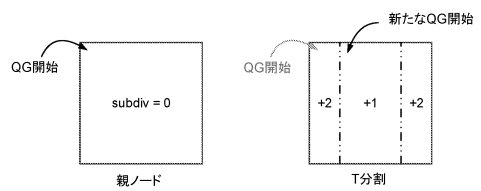
20

【図 20 E】



1/3 AB分割

【図 21】

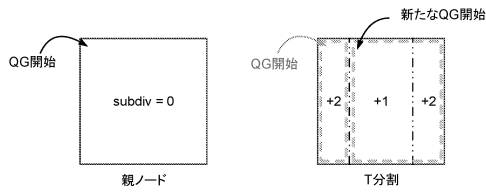


30

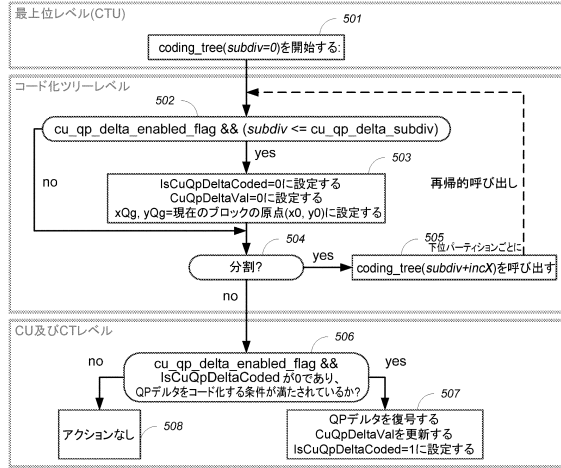
40

50

【 図 2 2 】



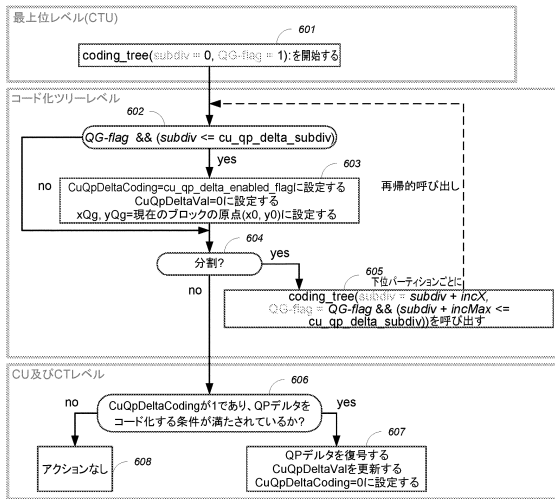
【 図 2 3 】



10

20

【 図 2 4 】



【 図 2 4 - 2 】

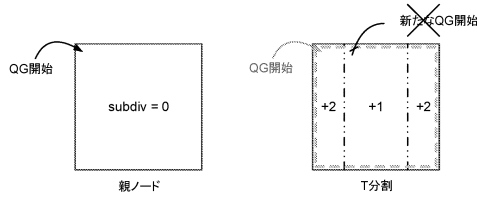
pic_parameter_set_rbsp( ) {	記述子
[...]	
cu_qp_delta_enabled_flag	u(1)
if (cu_qp_delta_enabled_flag)	
cu_qp_delta_subdiv	ue(v)
}	

30

40

50

【図 2 5】



【図 2 5 - 2】

	記述子
coding_tree_unit( ) {	
[...]	
if( slice_type == 1 && qtbt_dual_tree_intra_flag ) {	
dual_tree_implicit_qt_split( xCtb, yCtb, CtbLog2SizeY, 0, 0 )	
else	
coding_quadtree( xCtb, yCtb, CtbLog2SizeY, 0, 0, SINGLE_TREE )	
}	

10

【図 2 6】

	記述子
dual_tree_implicit_qt_split( x0, y0, log2CbSize, cqtDepth, cbSubdiv ) {	
if( cu_qp_delta_enabled_flag && cbSubdiv <= cu_qp_delta_subdiv ) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
xQg = x0	
yQg = y0	
}	
if( log2CbSize > 6 ) {	
x1 = x0 + ( 1 << ( log2CbSize - 1 ) )	
y1 = y0 + ( 1 << ( log2CbSize - 1 ) )	
dual_tree_implicit_qt_split( x0, y0, log2CbSize - 1, cqtDepth + 1, cbSubdiv + 2 )	
if( x1 < pic_width_in_luma_samples )	
dual_tree_implicit_qt_split( x1, y0, log2CbSize - 1, cqtDepth + 1, cbSubdiv + 2 )	
if( y1 < pic_height_in_luma_samples )	
dual_tree_implicit_qt_split( x0, y1, log2CbSize - 1, cqtDepth + 1, cbSubdiv + 2 )	
if( x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples )	
dual_tree_implicit_qt_split( x1, y1, log2CbSize - 1, cqtDepth + 1, cbSubdiv + 2 )	
} else {	
coding_quadtree( x0, y0, log2CbSize, cqtDepth, cbSubdiv, DUAL_TREE_LUMA )	
coding_quadtree( x0, y0, log2CbSize, cqtDepth, cbSubdiv, DUAL_TREE_CHROMA )	
}	
}	

【図 2 7】

	記述子
coding_quadtree( x0, y0, log2CbSize, cqtDepth, cbSubdiv, treeType ) {	
[...]	
if( cu_qp_delta_enabled_flag && cbSubdiv <= cu_qp_delta_subdiv ) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
xQg = x0	
yQg = y0	
}	
if( qt_split_cu_flag[ x0 ][ y0 ] ) {	
x1 = x0 + ( 1 << ( log2CbSize - 1 ) )	
y1 = y0 + ( 1 << ( log2CbSize - 1 ) )	
coding_quadtree( x0, y0, log2CbSize - 1, cqtDepth + 1, cbSubdiv + 2, treeType )	
if( x1 < pic_width_in_luma_samples )	
coding_quadtree( x1, y0, log2CbSize - 1, cqtDepth + 1, cbSubdiv + 2, treeType )	
if( y1 < pic_height_in_luma_samples )	
coding_quadtree( x0, y1, log2CbSize - 1, cqtDepth + 1, cbSubdiv + 2, treeType )	
if( x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples )	
coding_quadtree( x1, y1, log2CbSize - 1, cqtDepth + 1, cbSubdiv + 2, treeType )	
} else	
multi_type_tree( x0, y0, 1 << log2CbSize, 1 << log2CbSize, 0, 0, 0, cbSubdiv, treeType )	
}	

20

30

40

50

【 図 2 8 ( 1 ) 】

multi_type_tree(x0, y0, cbWidth, cbHeight, mttDepth, depthOffset, partIdx, cbSubdiv, treeType) {	記述子
if ( allowSplitBtVer    allowSplitBtHor    allowSplitTtVer    allowSplitTtHor ) && ( x0 + cbWidth <= pic_width_in_luma_samples ) && ( y0 + cbHeight <= pic_height_in_luma_samples ) )	
mtt_split_cu_flag	ae(v)
if ( cu_qp_delta_enabled_flag && cbSubdiv <= cu_qp_delta_subdiv ) {	
isCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
xQg = x0	
yQg = y0	
}	
if ( mtt_split_cu_flag ) {	
if ( ( allowSplitBtHor    allowSplitTtHor ) && ( allowSplitBtVer    allowSplitTtVer ) )	
mtt_split_cu_vertical_flag	ae(v)
if ( ( allowSplitBtVer && allowSplitTtVer && mtt_split_cu_vertical_flag )    ( allowSplitBtHor && allowSplitTtHor && !mtt_split_cu_vertical_flag ) )	
mtt_split_cu_binary_flag	ae(v)
if ( MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_VER ) {	
depthOffset += ( x0 + cbWidth > pic_width_in_luma_samples ) ? 1 : 0	
x1 = x0 + ( cbWidth / 2 )	
multi_type_tree(x0, y0, cbWidth / 2, cbHeight, mttDepth + 1, depthOffset, 0, cbSubdiv + 1, treeType)	
if ( x1 < pic_width_in_luma_samples )	
multi_type_tree(x1, y0, cbWidth / 2, cbHeight, mttDepth + 1, depthOffset, 1, cbSubdiv + 1, treeType)	
} else if ( MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_HOR ) {	
depthOffset += ( y0 + cbHeight > pic_height_in_luma_samples ) ? 1 : 0	
y1 = y0 + ( cbHeight / 2 )	
multi_type_tree(x0, y0, cbWidth, cbHeight / 2, mttDepth + 1, depthOffset, 0, cbSubdiv + 1, treeType)	
if ( y1 < pic_height_in_luma_samples )	
multi_type_tree(x0, y1, cbWidth, cbHeight / 2, mttDepth + 1, depthOffset, 1, cbSubdiv + 1, treeType)	
} else if ( MttSplitMode[x0][y0][mttDepth] == SPLIT_TT_VER ) {	
x1 = x0 + ( cbWidth / 4 )	

【 図 2 8 ( 2 ) 】

x2 = x0 + ( 3 * cbWidth / 4 )	
multi_type_tree(x0, y0, cbWidth / 4, cbHeight, mttDepth + 1, depthOffset, 0, cbSubdiv + 2, treeType)	
multi_type_tree(x1, y0, cbWidth / 2, cbHeight, mttDepth + 1, depthOffset, 1, cbSubdiv + 1, treeType)	
multi_type_tree(x2, y0, cbWidth / 4, cbHeight, mttDepth + 1, depthOffset, 2, cbSubdiv + 2, treeType)	
} else { /* SPLIT_TT_HOR */	
y1 = y0 + ( cbHeight / 4 )	
y2 = y0 + ( 3 * cbHeight / 4 )	
multi_type_tree(x0, y0, cbWidth, cbHeight / 4, mttDepth + 1, depthOffset, 0, cbSubdiv + 2, treeType)	
multi_type_tree(x0, y1, cbWidth, cbHeight / 2, mttDepth + 1, depthOffset, 1, cbSubdiv + 1, treeType)	
multi_type_tree(x0, y2, cbWidth, cbHeight / 4, mttDepth + 1, depthOffset, 2, cbSubdiv + 2, treeType)	
}	
} else	
coding_unit(x0, y0, cbWidth, cbHeight, treeType)	
}	

10

20

【 図 2 9 】

coding_quadtree(x0, y0, log2CbSize, cqtDepth, cbSubdiv, QG, treeType) {	記述子
[...]	
if ( QG && cbSubdiv <= cu_qp_delta_subdiv ) {	
CuQpDeltaCoding = cu_qp_delta_enabled_flag	
CuQpDeltaVal = 0	
xQg = x0	
yQg = y0	
}	
if ( qt_split_cu_flag[x0][y0] ) {	
x1 = x0 + ( 1 << ( log2CbSize - 1 ) )	
y1 = y0 + ( 1 << ( log2CbSize - 1 ) )	
coding_quadtree(x0, y0, log2CbSize - 1, cqtDepth + 1, cbSubdiv + 2, QG, treeType)	
if ( x1 < pic_width_in_luma_samples )	
coding_quadtree(x1, y0, log2CbSize - 1, cqtDepth + 1, cbSubdiv + 2, QG, treeType)	
if ( y1 < pic_height_in_luma_samples )	
coding_quadtree(x0, y1, log2CbSize - 1, cqtDepth + 1, cbSubdiv + 2, QG, treeType)	
if ( x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples )	
coding_quadtree(x1, y1, log2CbSize - 1, cqtDepth + 1, cbSubdiv + 2, QG, treeType)	
} else	
multi_type_tree(x0, y0, 1 << log2CbSize, 1 << log2CbSize, 0, 0, 0, cbSubdiv, QG, treeType)	
}	

【 図 3 0 ( 1 ) 】

multi_type_tree(x0, y0, cbWidth, cbHeight, mttDepth, depthOffset, partIdx, cbSubdiv, QG, treeType) {	記述子
if ( ( allowSplitBtVer    allowSplitBtHor    allowSplitTtVer    allowSplitTtHor ) && ( x0 + cbWidth <= pic_width_in_luma_samples ) && ( y0 + cbHeight <= pic_height_in_luma_samples ) )	
mtt_split_cu_flag	ae(v)
if ( QG && cbSubdiv <= cu_qp_delta_subdiv ) {	
CuQpDeltaCoding = cu_qp_delta_enabled_flag	
CuQpDeltaVal = 0	
xQg = x0	
yQg = y0	
}	
if ( mtt_split_cu_flag ) {	
if ( ( allowSplitBtHor    allowSplitTtHor ) && ( allowSplitBtVer    allowSplitTtVer ) )	
mtt_split_cu_vertical_flag	ae(v)
if ( ( allowSplitBtVer && allowSplitTtVer && mtt_split_cu_vertical_flag )    ( allowSplitBtHor && allowSplitTtHor && !mtt_split_cu_vertical_flag ) )	
mtt_split_cu_binary_flag	ae(v)
if ( MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_VER ) {	
depthOffset += ( x0 + cbWidth > pic_width_in_luma_samples ) ? 1 : 0	
x1 = x0 + ( cbWidth / 2 )	
multi_type_tree(x0, y0, cbWidth / 2, cbHeight, mttDepth + 1, depthOffset, 0, cbSubdiv + 1, QG, treeType)	
if ( x1 < pic_width_in_luma_samples )	
multi_type_tree(x1, y0, cbWidth / 2, cbHeight, mttDepth + 1, depthOffset, 1, cbSubdiv + 1, QG, treeType)	
} else if ( MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_HOR ) {	
depthOffset += ( y0 + cbHeight > pic_height_in_luma_samples ) ? 1 : 0	
y1 = y0 + ( cbHeight / 2 )	
multi_type_tree(x0, y0, cbWidth, cbHeight / 2, mttDepth + 1, depthOffset, 0, cbSubdiv + 1, QG, treeType)	
if ( y1 < pic_height_in_luma_samples )	
multi_type_tree(x0, y1, cbWidth, cbHeight / 2, mttDepth + 1, depthOffset, 1, cbSubdiv + 1, QG, treeType)	
} else if ( MttSplitMode[x0][y0][mttDepth] == SPLIT_TT_VER ) {	
x1 = x0 + ( cbWidth / 4 )	
x2 = x0 + ( 3 * cbWidth / 4 )	
xQg = QG && ( cbSubdiv + 2 <= cu_qp_delta_subdiv )	

30

40

50

【 図 3 0 ( 2 ) 】

multi_type_tree( x0, y0, cbWidth / 4, cbHeight, mttDepth + 1, depthOffset, 0, cbSubdiv + 2, sQG, treeType )	
multi_type_tree( x1, y0, cbWidth / 2, cbHeight, mttDepth + 1, depthOffset, 1, cbSubdiv + 1, sQG, treeType )	
multi_type_tree( x2, y0, cbWidth / 4, cbHeight, mttDepth + 1, depthOffset, 2, cbSubdiv + 2, sQG, treeType )	
} else { /* SPLIT_TT_HOR */	
y1 = y0 + ( cbHeight / 4 )	
y2 = y0 + ( 3 * cbHeight / 4 )	
sQG = QG && ( cbSubdiv + 2 <= cu_qp_delta_subdiv )	
multi_type_tree( x0, y0, cbWidth, cbHeight / 4, mttDepth + 1, depthOffset, 0, cbSubdiv + 2, sQG, treeType )	
multi_type_tree( x0, y1, cbWidth, cbHeight / 2, mttDepth + 1, depthOffset, 1, cbSubdiv + 1, sQG, treeType )	
multi_type_tree( x0, y2, cbWidth, cbHeight / 4, mttDepth + 1, depthOffset, 2, cbSubdiv + 2, sQG, treeType )	
}	
} else	
coding_unit( x0, y0, cbWidth, cbHeight, treeType )	
}	

【 図 3 1 】

transform_unit( x0, y0, tbWidth, tbHeight, treeType ) {	記述子
[...]	
if( [...] /* QP-delta coding conditions (e.g. cbf) */ ) {	
if( CuQpDeltaCoding ) {	
CuQpDeltaCoding = 0	
cu_qp_delta_abs	ae(v)
if( cp_qp_delta_abs )	
cu_qp_delta_sign_flag	ae(v)
}	
}	
[...]	
}	

10

【 図 3 2 】

coding_tree_unit() {	記述子
[...]	
if( slice_type = 1 && qtbt_dual_tree_intra_flag ) {	
dual_tree_implicit_qt_split( xCtb, yCtb, CtbLog2SizeY, 0, 0, 1, I )	
} else	
coding_quadtree( xCtb, yCtb, CtbLog2SizeY, 0, 0, 1, I, SINGLE_TREE )	
}	

【 図 3 3 】

coding_quadtree( x0, y0, log2CbSize, cqtDepth, cbSubdiv, QG, QGe, treeType ) {	記述子
[...]	
if( QG && cbSubdiv <= cu_qp_delta_subdiv ) {	
IsCuQpDeltaCoding = cu_qp_delta_enabled_flag	
CuQpDeltaVal = 0	
xQg = x0	
yQg = y0	
}	
if( QGe && cbSubdiv <= cu_chroma_qp_offset_subdiv )	
CuChromaQpOffsetCoding = cu_chroma_qp_offset_enabled_flag	
if( qt_split_cu_flag[ x0 ][ y0 ] ) {	
x1 = x0 + ( 1 << ( log2CbSize - 1 ) )	
y1 = y0 + ( 1 << ( log2CbSize - 1 ) )	
coding_quadtree( x0, y0, log2CbSize - 1, cqtDepth + 1, cbSubdiv + 2, QG, QGe, treeType )	
if( x1 < pic_width_in_luma_samples )	
coding_quadtree( x1, y0, log2CbSize - 1, cqtDepth + 1, ebSubdiv + 2, QG, QGe, treeType )	
if( y1 < pic_height_in_luma_samples )	
coding_quadtree( x0, y1, log2CbSize - 1, cqtDepth + 1, ebSubdiv + 2, QG, QGe, treeType )	
if( x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples )	
coding_quadtree( x1, y1, log2CbSize - 1, cqtDepth + 1, ebSubdiv + 2, QG, QGe, treeType )	
} else	
multi_type_tree( x0, y0, 1 << log2CbSize, 1 << log2CbSize, 0, 0, 0, ebSubdiv, QG, QGe, treeType )	
}	

20

30

40

50

【 図 3 4 】

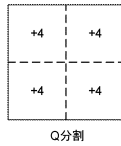
記述子	
multi_type_tree( x0, y0, cbWidth, cbHeight, mttDepth, depthOffset, partIdx, cbSubdiv, treeType ) {	
[...]	
} else { /* SPLIT_TT_HOR */	
y1 = y0 + ( cbHeight / 4 )	
y2 = y0 + ( 3 * cbHeight / 4 )	
sQG = QG && ( cbSubdiv + 2 <= cu_qp_delta_subdiv )	
sQGe = QGc && ( cbSubdiv + 2 <= cu_chroma_qp_offset_subdiv )	
multi_type_tree( x0, y0, cbWidth, cbHeight / 4, mttDepth + 1, depthOffset, 0, cbSubdiv + 2, sQG, sQGe, treeType )	
multi_type_tree( x0, y1, cbWidth, cbHeight / 2, mttDepth + 1, depthOffset, 1, cbSubdiv + 1, sQG, sQGe, treeType )	
multi_type_tree( x0, y2, cbWidth, cbHeight / 4, mttDepth + 1, depthOffset, 2, cbSubdiv + 2, sQG, sQGe, treeType )	
}	
} else	
coding_unit( x0, y0, cbWidth, cbHeight, treeType )	
}	

【 図 3 5 】

記述子	
transform_unit( x0, y0, tbWidth, tbHeight, treeType ) {	
[...]	
if( [...] /* QP-delta coding conditions (e.g. cbf) */ ) {	
if( CuQpDeltaCoding ) {	
CuQpDeltaCoding = 0	
cu_qp_delta_abs	ae(v)
if( cp_qp_delta_abs )	
cu_qp_delta_sign_flag	ae(v)
}	
if( [...] /* chroma QP offset coding conditions (e.g. cbf) */	
&& CuChromaQpOffsetCoding ) {	
cu_chroma_qp_offset_flag	ae(v)
if( cu_chroma_qp_offset_flag && chroma_qp_offset_list_len_minus1 > 0 )	
cu_chroma_qp_offset_idx	ae(v)
}	
}	
[...]	
}	

10

【 図 1 8 8 A 】



20

30

40

50

## フロントページの続き

(33)優先権主張国・地域又は機関

欧州特許庁(EP)  
弁理士 阿部 豊隆

(72)発明者 ド ラグランジュ, フィリップ

フランス国, 3 5 5 7 6 セゾン - セビニエ, ザック デ シャン ブラン, アベニュー デ シャン  
ブラン 9 7 5, テクニカラー・アール・アンド・ディー フランス

(72)発明者 ボルデ, フィリップ

フランス国, 3 5 5 7 6 セゾン - セビニエ, ザック デ シャン ブラン, アベニュー デ シャン  
ブラン 9 7 5, テクニカラー・アール・アンド・ディー フランス

(72)発明者 フランソワ, エドワール

フランス国, 3 5 5 7 6 セゾン - セビニエ, ザック デ シャン ブラン, アベニュー デ シャン  
ブラン 9 7 5, テクニカラー・アール・アンド・ディー フランス

審査官 鉢呂 健

(56)参考文献 国際公開第 2 0 1 9 / 1 3 5 6 0 1 ( W O , A 1 )

ZHAO, Yin et al. , Quantization parameter signaling , Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 12th Meeting: Macao, CN, 3-12 Oct. 2018, [JVET-L0362-v3], JVET-L0362 (version 3), ITU-T, 2018年10月06日, <URL:https://jvet-experts.org/doc\_end\_user/documents/12\_Macao/wg11/JVET-L0362-v3.zip>: JVET-L0362-v3.docx: pp.1-11

DE LAGRANGE, P et al. , CE7-related: Quantization Group size uniformity , Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 13th Meeting: Marrakech, MA, 9-18 Jan. 2019, [JVET-M0113], JVET-M0113 (version 1), ITU-T, 2018年12月30日, <URL:https://jvet-experts.org/doc\_end\_user/documents/13\_Marrakech/wg11/JVET-M0113-v1.zip>: JVET-M0113.docx: pp.1-8

(58)調査した分野 (Int.Cl., D B 名)

H 0 4 N 1 9 / 0 0 - 1 9 / 9 8