

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7359951号
(P7359951)

(45)発行日 令和5年10月11日(2023.10.11)

(24)登録日 令和5年10月2日(2023.10.2)

(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 19/593 (2014.01)

H 0 4 N 19/593

請求項の数 12 (全55頁)

(21)出願番号	特願2022-519449(P2022-519449)	(73)特許権者	520353802
(86)(22)出願日	令和3年4月6日(2021.4.6)		テンセント・アメリカ・エルエルシー
(65)公表番号	特表2022-549910(P2022-549910 A)		アメリカ合衆国 9 4 3 0 6 カリフォルニア州 パロアルト パーク・ブルヴァード 2 7 4 7
(43)公表日	令和4年11月29日(2022.11.29)	(74)代理人	100107766
(86)国際出願番号	PCT/US2021/025870		弁理士 伊東 忠重
(87)国際公開番号	WO2021/207124	(74)代理人	100070150
(87)国際公開日	令和3年10月14日(2021.10.14)		弁理士 伊東 忠彦
審査請求日	令和4年3月28日(2022.3.28)	(74)代理人	100135079
(31)優先権主張番号	63/007,187		弁理士 宮崎 修
(32)優先日	令和2年4月8日(2020.4.8)	(72)発明者	リ, リン
(33)優先権主張国・地域又は機関	米国(US)		アメリカ合衆国 9 4 3 0 6 カリフォルニア州 パロアルト パーク・ブルバード 2 7 4 7 テンセント アメリカ エル
(31)優先権主張番号	63/029,000		最終頁に続く
(32)優先日	令和2年5月22日(2020.5.22)		
	最終頁に続く		

(54)【発明の名称】 ビデオ符号化のための方法、装置、媒体およびコンピュータ・プログラム

(57)【特許請求の範囲】

【請求項 1】

デコーダによって実行される、ビデオ・デコードの方法であって：

符号化ビデオ・ビットストリームの予測情報における複数の画像スライスについてのプロファイル情報をデコードする段階であって、前記プロファイル情報は、前記符号化ビデオ・ビットストリーム内の前記画像スライスのそれぞれがイントラ符号化されるプロファイルのプロファイル識別情報を含み、前記プロファイル情報は前記符号化ビデオ・ビットストリーム内の各画像スライスが1つのピクチャーに含まれているかどうかを示す情報を含み、該情報が前記符号化ビデオ・ビットストリーム内の各画像スライスが1つのピクチャーに含まれることを示すことが、前記符号化ビデオ・ビットストリーム内の各画像スライスがイントラ符号化されていることを示す、段階と；

前記符号化ビデオ・ビットストリーム内の各画像スライスに対してイントラ予測を実行する段階と；

少なくとも1つの画像ピクチャーを、前記イントラ予測に基づいて再構成する段階とを含む、方法。

【請求項 2】

前記プロファイル情報は、前記符号化ビデオ・ビットストリーム内の各画像スライスがイントラ符号化されているかどうかを示す第1のフラグをさらに含む、請求項 1 に記載の方法。

【請求項 3】

前記第1のフラグは、前記符号化ビデオ・ビットストリーム内の各画像スライスが1つのピクチャーに含まれているかどうかを示す前記情報の後にデコードされる、請求項 2 に記載の方法。

【請求項 4】

前記第1のフラグは、前記符号化ビデオ・ビットストリーム内の各画像スライスがイントラ符号化される前記プロファイルの前記プロファイル識別情報に基づいて、前記符号化ビデオ・ビットストリーム内の各画像スライスがイントラ符号化されていることを示す、請求項 2 に記載の方法。

【請求項 5】

前記符号化ビデオ・ビットストリーム内の各画像スライスが1つのピクチャーに含まれているかどうかを示す前記情報は、前記プロファイルが前記符号化ビデオ・ビットストリーム内に1つのピクチャーのみが含まれる静止ピクチャー・プロファイルであることに基づいて、前記符号化ビデオ・ビットストリーム内の各画像スライスが1つのピクチャーに含まれることを示す、請求項 2 に記載の方法。

【請求項 6】

非イントラ関連構文要素は、(i) 前記第1のフラグが、前記符号化ビデオ・ビットストリーム内の各画像スライスがイントラ符号化されることを示すこと、(ii) 前記符号化ビデオ・ビットストリーム内の各画像スライスが1つのピクチャーに含まれているかどうかを示す前記情報が、前記符号化ビデオ・ビットストリーム内の各画像スライスが1つのピクチャーに含まれることを示すことのうちの1つに基づいて、前記予測情報に含まれない、請求項 2 に記載の方法。

【請求項 7】

前記予測情報は、前記符号化ビデオ・ビットストリーム内の各画像スライスがイントラ符号化され、1つのピクチャーに含まれるかどうかを示す第3のフラグを含み、前記第3のフラグは、前記プロファイル情報には含まれない、請求項 1 ないし 6 のうちいずれか一項に記載の方法。

【請求項 8】

前記第3のフラグは、前記符号化ビデオ・ビットストリーム内の各画像スライスが1つのピクチャーに含まれているかどうかを示す前記情報が、前記符号化ビデオ・ビットストリーム内の各画像スライスが1つのピクチャーに含まれることを示すことに基づいて、前記符号化ビデオ・ビットストリーム内の各画像スライスがイントラ符号化され、1つのピクチャーに含まれることを示す、請求項 7 が請求項 2 ないし 6 のうちいずれか一項を引用する場合の請求項 7 に記載の方法。

【請求項 9】

前記符号化ビデオ・ビットストリーム内の各画像スライスが1つのピクチャーに含まれているかどうかを示す前記情報は、Main 10 静止ピクチャー・プロファイルおよびMain 4:4:4 10 静止ピクチャー・プロファイルのうち的一方であり、

当該方法は、Main 10 静止ピクチャー・プロファイルおよびMain 4:4:4 10 静止ピクチャー・プロファイルのうちの方に準拠する前記複数の画像スライスに対応する前記符号化ビデオ・ビットストリームに、1つのピクチャーのみが含まれるように制約することをさらに含む、請求項 1 ないし 8 のうちいずれか一項に記載の方法。

【請求項 10】

請求項 1 ないし 9 のうちいずれか一項に記載の方法を実行するように構成された処理回路を有する、装置。

【請求項 11】

少なくとも1つのプロセッサによって実行されたときに該少なくとも1つのプロセッサに請求項 1 ないし 9 のうちいずれか一項に記載の方法を実行させる命令を記憶している非一時的なコンピュータ読み取り可能な記憶媒体。

【請求項 12】

少なくとも1つのプロセッサに請求項 1 ないし 9 のうちいずれか一項に記載の方法を実行させるためのコンピュータ・プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

参照による組み込み

本願は、2021年4月1日出願された米国特許出願第17/220,481号「ビデオ符号化のための方法および装置」の優先権の利益を主張し、同出願は2020年4月8日出願された米国仮特許出願第63/007,187号「さまざまなプロファイルのための構文要素に対する制約条件」および2020年5月22日出願された米国仮特許出願第63/029,000号「一般制約条件フラグのグループ」の優先権の利益を主張する。これら先の出願の開示全体は、その全体が参照により本明細書に組み込まれる。

【0002】

技術分野

本開示は、ビデオ符号化に一般的に関連する実施形態を記載する。

【背景技術】

【0003】

本明細書で提供される背景説明は、本開示の文脈を概括的に提示するためのものである。本願で名前が挙がっている発明者の仕事であってその仕事がこの背景セクションに記載されている範囲におけるもの、また、他の意味で出願時に先行技術として適格でないことがありうる本記述の諸側面は、明示的にも暗黙的にも本開示に対する先行技術として認められない。

【0004】

ビデオ符号化および復号は、動き補償を伴うピクチャー間予測を用いて実行できる。非圧縮デジタル・ビデオは、一連のピクチャーを含むことができ、各ピクチャーは、たとえば1920×1080のルミナンス・サンプルおよび関連するクロミナンス・サンプルの空間的寸法を有する。一連のピクチャーは、固定または可変のピクチャー・レート（非公式にはフレーム・レートとしても知られる）、たとえば、60ピクチャー毎秒または60Hzを有することができる。非圧縮ビデオは、かなりのビットレート要件を有する。たとえば、サンプル当たり8ビットの1080p60 4:2:0ビデオ（60Hzのフレーム・レートでの1920×1080のルミナンス・サンプル解像度）は、1.5Gbit/sに近い帯域幅を必要とする。そのようなビデオの1時間は、600Gバイトを超える記憶スペースを必要とする。

【0005】

ビデオ符号化および復号の1つの目的は、圧縮による入力ビデオ信号の冗長性の低減でありうる。圧縮は、前述の帯域幅または記憶スペースの要求を、場合によっては2桁以上も低減するのに役立つことがある。可逆圧縮および不可逆圧縮の両方、ならびにそれらの組み合わせを用いることができる。可逆圧縮とは、圧縮されたもとの信号から、もとの信号の正確なコピーが再構成できる技術をいう。不可逆圧縮を使用する場合、再構成された信号は、もとの信号と同一ではないことがありうるが、もとの信号と再構成された信号との間の歪みは、再構成された信号を意図された用途のために有用にするのに十分小さい。ビデオの場合、不可逆圧縮が広く用いられている。許容される歪みの量はアプリケーションに依存し、たとえば、ある種の消費者ストリーミングアプリケーションのユーザーは、テレビ配信アプリケーションのユーザーよりも高い歪みを許容することがある。達成可能な圧縮比は、より高い許容可能／認容可能な歪みはより高い圧縮比をもたらすことができる、ということを反映できる。

【0006】

ビデオ・エンコーダおよびデコーダは、たとえば動き補償、変換、量子化、およびエントロピー符号化を含むいくつかの広範なカテゴリーからの技術を利用することができる。

【0007】

ビデオ・コーデック技術は、イントラ符号化として知られる技術を含むことができる。

イントラ符号化では、サンプル値は、以前に再構成された参照ピクチャーからのサンプルまたは他のデータを参照することなく表現される。いくつかのビデオ・コーデックでは、ピクチャーは空間的にサンプルのブロックに分割される。サンプルのすべてのブロックがイントラモードで符号化される場合、そのピクチャーはイントラ・ピクチャーでありうる。イントラ・ピクチャーと、独立デコーダ・リフレッシュ・ピクチャーのようなその派生物は、デコーダ状態をリセットするために使用でき、よって、符号化ビデオ・ビットストリームおよびビデオセッションにおける最初のピクチャーとして、または静止画像として使用できる。イントラ・ブロックのサンプルを変換にかけることができ、変換係数は、エントロピー符号化の前に量子化されることができる。イントラ予測は、変換前領域におけるサンプル値を最小化する技術でありうる。場合によっては、変換後のDC値が小さく、AC係数が小さいほど、エントロピー符号化後のブロックを表わすために所与の量子化ステップサイズで必要とされるビット数が少なくなる。

10

【0008】

たとえばMPEG-2世代の符号化技術から知られているような伝統的なイントラ符号化は、イントラ予測を使用しない。しかしながら、いくつかのより新しいビデオ圧縮技術は、たとえば、空間的に近傍であり、デコード順で先行するデータのブロックのエンコード/デコード中に得られた周囲のサンプル・データおよび/またはメタデータから試みる技術を含む。そのような技法は、以下では「イントラ予測」技法と呼ばれる。少なくともいくつかの場合には、イントラ予測は再構成中の現在ピクチャーからの参照データのみを使用し、参照ピクチャーからの参照データは使用しないことに注意されたい。

20

【0009】

さまざまな形のイントラ予測がありうる。所与のビデオ符号化技術において、そのような技法の二つ以上が使用できる場合、使用される技法は、イントラ予測モードで符号化されることができる。ある種の場合には、モードは、サブモードおよび/またはパラメータを有することができ、それらは、個別に符号化されることができ、またはモード符号語に含められることができる。所与のモード/サブモード/パラメータの組み合わせのためにどの符号語を使用するかは、イントラ予測を通して符号化効率利得に影響を与える可能性があり、符号語をビットストリームに変換するために使用されるエントロピー符号化技術も同様に影響を与える可能性がある。

【0010】

30

イントラ予測のあるモードがH.264で導入され、H.265で洗練され、共同探査モデル(JEM)、多用途ビデオ符号化(VVC)、およびベンチマークセット(BMS)のようなより新しい符号化技術においてさらに洗練された。予測子ブロックは、すでに利用可能なサンプルに属する近傍サンプル値を使用して形成されることができる。近傍サンプルのサンプル値が、ある方向に従って予測子ブロックにコピーされる。使用される方向への参照は、ビットストリームにおいて符号化されることができ、またはそれ自身予測されてもよい。

【0011】

図1Aを参照すると、右下に、H.265の33個の可能な予測子方向(35個のイントラモードのうち33個の角度モードに対応する)から知られている9個の予測子方向のサブセットが描かれている。矢印が収束する点(101)は、予測されるサンプルを表わす。矢印は、サンプルが予測される方向を表わす。たとえば、矢印(102)は、サンプル(101)が、水平から45度の角度の右上のサンプル(単数または複数)から予測されることを示す。同様に、矢印(103)は、サンプル(101)が、水平から22.5度の角度の、サンプル(101)の左下のサンプル(単数または複数)から予測されることを示す。

40

【0012】

引き続き図1Aを参照すると、左上には、4×4サンプルの正方形ブロック(104)が描かれている(太い破線で示されている)。正方形ブロック(104)は、16個のサンプルを含み、各サンプルは「S」とY次元におけるその位置(たとえば、行インデックス)およびX次元におけるその位置(たとえば、列インデックス)でラベル付けされている。たとえば、サンプルS21は、Y次元の(上から)2番目のサンプルであり、X次元の(左から)最

50

初のサンプルである。同様に、サンプルS44は、YおよびX次元の両方においてブロック（104）内の第4のサンプルである。ブロックが4×4サンプルのサイズなので、S44は右下にある。さらに、同様の番号付けスキームに従う参照サンプルが示されている。参照サンプルは、Rと、ブロック（104）に対するそのY位置（たとえば、行インデックス）およびX位置（列インデックス）でラベル付けされる。H.264とH.265の両方において、予測サンプルは再構成中のブロックの近傍であり、そのため負の値を使用する必要はない。

【0013】

ピクチャー内予測は、信号伝達される予測方向によって充当される近傍サンプルから参照サンプル値をコピーすることによって機能できる。たとえば、符号化ビデオ・ビットストリームは、このブロックについて、矢印（102）と整合する予測方向を示す信号伝達を含むと想定する。すなわち、サンプルは、水平から45度の角度の右上の予測サンプル（単数または複数）から予測される。その場合、サンプルS41、S32、S23、およびS14は、同じ参照サンプルR05から予測される。次いで、サンプルS44は、参照サンプルR08から予測される。

【0014】

ある種の場合には、特に方向が45度で割り切れない場合には、参照サンプルを計算するために、複数の参照サンプルの値が、たとえば補間によって組み合わせることができる。

【0015】

ビデオ符号化技術の発達に伴い、可能な方向の数が増加してきた。H.264（2003年）では、9つの異なる方向が表現できた。これは、H.265（2013年）では33に増加し、本開示の時点でのJEM/VVC/BMSは、最大65の方向をサポートできる。最も可能性の高い方向を同定するために実験が行われ、より可能性の低い方向についてのあるペナルティを受け入れつつ、それらの可能性の高い方向を少数のビットで表現するために、エントロピー符号化におけるある種の技法が使用される。さらに、方向自身が、近傍のすでにデコードされたブロックで使用された近傍方向から予測できることがある。

【0016】

図1Bは、時間とともに増加する予測方向の数を示すために、JEMによる65個のイントラ予測方向を描く概略図（105）を示している。

【0017】

方向を表わす符号化ビデオ・ビットストリームにおけるイントラ予測方向ビットのマッピングは、ビデオ符号化技術ごとに異なることができ、たとえば、予測方向のイントラ予測モードへの単純な直接的マッピングから、符号語、最確モードに関わる複雑な適応方式、および同様の技法までありうる。しかしながら、どの場合でも、ビデオコンテンツにおいて、他のある種の方向よりも統計的に起こりにくいある種の方向が存在しうる。ビデオ圧縮の目標は冗長性の低減であるので、良好に機能するビデオ符号化技術においては、そうしたより可能性の低い方法は、より可能性の高い方向よりもより多くのビット数によって表わされる。

【0018】

動き補償は、不可逆圧縮技法であることがあり、かつ、以前に再構成されたピクチャーまたはその一部（参照ピクチャー）からのサンプル・データのブロックが、動きベクトル（以下、MV）によって示される方向に空間的にシフトされた後に、新しく再構成されるピクチャーまたはその一部の予測のために使用される技法に関することがある。場合によっては、参照ピクチャーは、現在再構成中のピクチャーと同じであることもできる。MVは、XおよびYの2次元、または3次元を有することができ、第3の次元は、使用される参照ピクチャーの指示である（これは、間接的に、時間次元でありうる）。

【0019】

いくつかのビデオ圧縮技術では、サンプル・データのある領域に適用可能なMVは、他のMVから、たとえば、再構成中の領域に空間的に隣接し、デコード順でそのMVに先行するサンプル・データの別の領域に関連するMVから予測されることがある。そうすることにより、MVの符号化に必要とされるデータ量を大幅に削減することができ、それによ

10

20

30

40

50

り冗長性を除去し、圧縮を増加させることができる。MV予測が有向に機能できるのは、たとえば、カメラから導出される入力ビデオ信号（ナチュラルビデオとして知られる）を符号化する際に、単一のMVが適用可能である領域よりも大きな領域が同様の方向に移動し、よって、ある種の場合には、近傍領域のMVから導出された同様のMVを用いて予測できるという、統計的確からしさがあるからである。その結果、所与の領域について見出されるMVが、周囲のMVから予測されるMVと同様または同一であることになり、そして、それは、エントロピー符号化の後、MVを直接符号化する場合に使用されるであろうものよりも少数のビットで表現できる。いくつかの場合には、MV予測は、もとの信号（すなわち、サンプルストリーム）から導出された信号（すなわち、MV）の可逆圧縮の例でありうる。他の場合には、MV予測自身が、たとえば、いくつかの周囲のMVから予測子を計算する際の丸め誤差のために、不可逆であることがある。

10

【0020】

H.265/HEVC (ITU-T Rec. H.265、「高効率ビデオ符号化」(High Efficiency Video Coding)、December 2016)には、さまざまなMV予測機構が記述されている。H.265が提供する多くのMV予測機構のうち、本明細書では、以後、「空間マージ (spatial merge)」と呼ばれる技法が記載される。

【0021】

図1Cを参照すると、現在ブロック(101)は、空間的にシフトされた同じサイズの前のブロックから予測可能であることが、動き探索プロセスの間にエンコードによって見出されたサンプルを含むことができる。そのMVを直接符号化する代わりに、MVは、一つまたは複数の参照ピクチャーに関連付けられたメタデータから、たとえば(デコード順で)最新の参照ピクチャーから、A0、A1、およびB0、B1、B2(それぞれ112ないし116)と記される5つの周囲のサンプルのいずれかに関連付けられたMVを使用して、導出できる。H.265では、MV予測は、近傍ブロックが使用しているのと同じ参照ピクチャーからの予測子を使用することができる。

20

【発明の概要】

【課題を解決するための手段】

【0022】

本開示の諸側面は、ビデオ・エンコード/デコードのための装置を提供する。装置は、符号化ビデオ・ビットストリームの予測情報における複数の画像スライスのプロファイル情報をデコードする処理回路を含む。プロファイル情報は、符号化されたビデオ・ビットストリーム内の画像スライスのそれぞれがイントラ符号化されるプロファイルのプロファイル識別情報を含む。処理回路は、符号化ビデオ・ビットストリーム内の各画像スライスに対してイントラ予測を実行する。処理回路は、イントラ予測に基づいて少なくとも1つの画像ピクチャーを再構成する。

30

【0023】

ある実施形態では、プロファイル情報は、符号化ビデオ・ビットストリーム内の各画像スライスがイントラ符号化されているかどうかを示す第1のフラグと、符号化ビデオ・ビットストリーム内の各画像スライスが1つのピクチャーに含まれているかどうかを示す第2のフラグとを含む。

40

【0024】

ある実施形態では、第1のフラグは、第2のフラグの後にデコードされ、第2のフラグが符号化ビデオ・ビットストリーム内の各画像スライスが1つのピクチャーに含まれることを示すことに基づいて、符号化ビデオ・ビットストリーム内の各画像スライスがイントラ符号化されていることを示す。

【0025】

ある実施形態では、第1のフラグは、符号化ビデオ・ビットストリーム内の各画像スライスがイントラ符号化されるプロファイルのプロファイル識別情報に基づいて、符号化ビデオ・ビットストリーム内の各画像スライスがイントラ符号化されていることを示す。

【0026】

50

ある実施形態では、第2のフラグは、前記プロファイルが符号化ビデオ・ビットストリーム内に1つのピクチャーのみが含まれる静止ピクチャー・プロファイルであることに基
づいて、符号化ビデオ・ビットストリーム内の各画像スライスが1つのピクチャーに含ま
れることを示す。

【0027】

ある実施形態では、非イントラ関連構文要素は、(i)第1のフラグが、符号化ビデオ・
ビットストリーム内の各画像スライスがイントラ符号化されることを示すこと、(ii)第2
のフラグが、符号化ビデオ・ビットストリーム内の各画像スライスが1つのピクチャーに
含まれることを示すことのうちの1つに基づいて、前記予測情報に含まれない。

【0028】

ある実施形態では、前記予測情報は、符号化ビデオ・ビットストリーム内の各画像スラ
イスがイントラ符号化され、1つのピクチャーに含まれるかどうかを示す第3のフラグを含
む。第3のフラグは、前記プロファイル情報には含まれない。

【0029】

ある実施形態では、第3のフラグは、第2のフラグが、符号化ビデオ・ビットストリー
ム内の各画像スライスが1つのピクチャーに含まれることを示すことに基づいて、符号化ビ
デオ・ビットストリーム内の各画像スライスがイントラ符号化され、1つのピクチャーに
含まれることを示す。

【0030】

本開示の諸側面は、ビデオ・エンコード/デコードのための方法を提供する。この方法
では、符号化ビデオ・ビットストリームの予測情報における複数の画像スライスのプロフ
ァイル情報がデコードされる。プロファイル情報は、符号化ビデオ・ビットストリーム内
の画像スライスのそれぞれがイントラ符号化されるプロファイルのプロファイル識別情報
を含む。符号化ビデオ・ビットストリーム内の各画像スライスのイントラ予測が実行され
る。少なくとも1つの画像ピクチャーが、前記イントラ予測に基づいて再構成される。

【0031】

本開示の諸側面は、ビデオ・エンコード/デコードのための装置を提供する。装置は、
符号化ビデオ・ビットストリームの予測情報におけるプロファイル情報をデコードする処
理回路を含む。プロファイル情報は、構文要素の複数のグループを含み、符号化ビデオ・
ビットストリームについてのプロファイルを示す。処理回路は、プロファイル情報におい
て示されたプロファイルに基づいて、構文要素の前記複数のグループのうちの少なくと
も1つを決定する。処理回路は、構文要素の前記複数のグループのうちの前記決定された少
なくとも1つに基づいて、前記予測情報に含まれる構文要素をデコードする。処理回路は
、前記予測情報に含まれるデコードされた構文要素に基づいて、少なくとも1つのピク
チャーを再構成する。

【0032】

ある実施形態では、前記プロファイルについての構文要素の前記複数のグループのう
ちの前記決定された少なくとも1つの順序は、前記プロファイル情報における構文要素の前
記複数のグループの所定の順序に従う。

【0033】

ある実施形態では、前記プロファイル情報における構文要素の前記複数のグループのそ
れぞれについて、バイト整列がチェックされる。

【0034】

本開示の諸側面は、ビデオ・エンコード/デコードのための方法を提供する。この方法
では、符号化ビデオ・ビットストリームの予測情報におけるプロファイル情報がデコード
する。プロファイル情報は、構文要素の複数のグループを含み、符号化ビデオ・ビットス
トリームについてのプロファイルを示す。構文要素の前記複数のグループのうちの少なく
とも1つは、前記プロファイル情報において示される前記プロファイルに基づいて決定さ
れる。前記予測情報に含まれる構文要素は、構文要素の前記複数のグループのうちの前記
決定された少なくとも1つに基づいてデコードされる。前記予測情報におけるデコードさ

10

20

30

40

50

れた構文要素に基づいて、少なくとも1つのピクチャーが再構成される。

【0035】

本開示の諸側面は、ビデオ・デコードのためにコンピュータによって実行されたときに、該コンピュータに、ビデオ・デコードのための前記の諸方法のいずれか1つまたは組み合わせを実行させる命令を記憶している非一時的なコンピュータ読み取り可能媒体をも提供する。

【図面の簡単な説明】

【0036】

開示された主題のさらなる特徴、性質、およびさまざまな利点は、以下の詳細な説明および添付の図面からより明白になるであろう。

10

【0037】

【図1A】イントラ予測モードの例示的なサブセットの概略図である。

【0038】

【図1B】例示的なイントラ予測方向の説明図である。

【0039】

【図1C】一例における現在ブロックおよびその周囲の空間的マージ候補の概略図である。

【0040】

【図2】ある実施形態による通信システムの簡略化されたブロック図の概略図である。

【0041】

【図3】ある実施形態による通信システムの簡略化されたブロック図の概略図である。

20

【0042】

【図4】ある実施形態によるデコーダの簡略化されたブロック図の概略図である。

【0043】

【図5】ある実施形態によるエンコーダの簡略化されたブロック図の概略図である。

【0044】

【図6】別の実施形態によるエンコーダのブロック図を示す。

【0045】

【図7】別の実施形態によるデコーダのブロック図を示す。

【0046】

【図8】ある実施形態による例示的なフローチャートを示す。

30

【0047】

【図9】ある実施形態による例示的なフローチャートを示す。

【0048】

【図10】ある実施形態によるコンピュータ・システムの概略図である。

【発明を実施するための形態】

【0049】

1. ビデオ・デコーダおよびエンコーダ・システム

【0050】

図2は、本開示のある実施形態による通信システム(200)の簡略化されたブロック図を示す。通信システム(200)は、たとえばネットワーク(250)を介して互いに通信することができる複数の端末装置を含む。たとえば、通信システム(200)は、ネットワーク(250)を介して相互接続された第1の対の端末装置(210)および(220)を含む。図2の例では、第1の対の端末装置(210)および(220)は、データの一方向伝送を実行する。たとえば、端末装置(210)は、ネットワーク(250)を介した他方の端末装置(220)への伝送のために、ビデオ・データ(たとえば、端末装置(210)によって捕捉されたビデオ・ピクチャーのストリーム)を符号化してもよい。エンコードされたビデオ・データは、一つまたは複数の符号化ビデオ・ビットストリームの形で伝送されることができる。端末装置(220)は、ネットワーク(250)から、符号化ビデオ・データを受信し、符号化ビデオ・データをデコードしてビデオ・ピクチャーを復元し、復元されたビデオ・データに従ってビデオ・ピクチャーを表示してもよい。一方向データ伝送は、メディア

40

50

・サービス・アプリケーション等において一般的でありうる。

【0051】

別の例では、通信システム(200)は、たとえばビデオ会議中に発生しうる符号化されたビデオ・データの双方向伝送を実行する第2の対の端末装置(230)および(240)を含む。データの双方向伝送のために、一例では、端末装置(230)および(240)の各端末装置は、ネットワーク(250)を介した、端末装置(230)および(240)のうちの他方の端末装置への伝送のために、ビデオ・データ(たとえば、端末装置によって捕捉されたビデオ・ピクチャーのストリーム)を符号化してもよい。端末装置(230)および(240)の各端末装置は、端末装置(230)および(240)のうちの他方の端末装置によって送信された符号化されたビデオ・データを受信してもよく、符号化されたビデオ・データをデコードして、ビデオ・ピクチャーを復元し、復元されたビデオ・データに従って、アクセス可能な表示装置においてビデオ・ピクチャーを表示してもよい。

10

【0052】

図2の例では、端末装置(210)、(220)、(230)および(240)は、サーバー、パーソナルコンピュータおよびスマートフォンとして示されてもよいが、本開示の原理は、それに限定されなくてもよい。本開示の実施形態は、ラップトップ・コンピュータ、タブレット・コンピュータ、メディア・プレーヤー、および/または専用のビデオ会議設備での応用を見出す。ネットワーク(250)は、たとえば有線(ワイヤード)および/または無線[ワイヤレス]通信ネットワークを含む、端末装置(210)、(220)、(230)および(240)の間で符号化されたビデオ・データを伝達する任意の数のネットワークを表わす。通信ネットワーク(250)は、回線交換および/またはパケット交換チャネルにおいてデータを交換してもよい。代表的なネットワークは、遠隔通信ネットワーク、ローカルエリアネットワーク、ワイドエリアネットワークおよび/またはインターネットを含む。今の議論の目的のために、ネットワーク(250)のアーキテクチャーおよびトポロジーは、以下に説明しない限り、本開示の動作には重要ではないことがある。

20

【0053】

図3は、開示される主題のためのアプリケーションの例として、ストリーミング環境におけるビデオ・エンコーダおよびビデオ・デコーダの配置を示す。開示される主題は、たとえば、ビデオ会議、デジタルTV、CD、DVD、メモリスティックなどを含むデジタル媒体上の圧縮ビデオの記憶などを含む、他のビデオ対応アプリケーションにも等しく適用可能でありうる。

30

【0054】

ストリーミング・システムは、ビデオ源(301)、たとえばデジタル・カメラを含むことができ、たとえば非圧縮のビデオ・ピクチャーのストリーム(302)を生成する捕捉サブシステム(313)を含んでいてもよい。一例では、ビデオ・ピクチャーのストリーム(302)は、デジタル・カメラによって取り込まれたサンプルを含む。エンコードされたビデオ・データ(304)(または符号化されたビデオ・ビットストリーム)と比較した場合の低いデータ・ボリュームを強調するために太線として描かれているビデオ・ピクチャーのストリーム(302)は、ビデオ源(301)に結合されたビデオ・エンコーダ(303)を含む電子装置(320)によって処理されることができる。ビデオ・エンコーダ(303)は、以下により詳細に説明されるように、開示される主題の諸側面を可能にし、または実現するためのハードウェア、ソフトウェア、またはそれらの組み合わせを含むことができる。ビデオ・ピクチャーのストリーム(302)と比較した場合の、より低いデータ・ボリュームを強調するために細い線として描かれている、エンコードされたビデオ・データ(304)(またはエンコードされたビデオ・ビットストリーム(304))は、将来の使用のためにストリーミング・サーバー(305)に記憶されることができる。図3のクライアント・サブシステム(306)および(308)のような一つまたは複数のストリーミング・クライアント・サブシステムは、ストリーミング・サーバー(305)にアクセスして、エンコードされたビデオ・データ(304)のコピー(307)および(309)を取り出すことができる。クライアント・サブシステム(306)は、たとえば電子装置(330)内にビデオ・

40

50

デコーダ(310)を含むことができる。ビデオ・デコーダ(310)は、エンコードされたビデオ・データの到来コピー(307)をデコードし、ディスプレイ(312)(たとえば表示画面)または他のレンダリング装置(図示せず)上にレンダリングできるビデオ・ピクチャーの出ていくストリーム(311)を生成する。いくつかのストリーミング・システムでは、エンコードされたビデオ・データ(304)、(307)、および(309)(たとえば、ビデオ・ビットストリーム)は、ある種のビデオ符号化/圧縮標準に従ってエンコードされることができる。これらの標準の例は、ITU-T勧告H.265を含む。一例では、開発中のビデオ符号化規格は、非公式にVVCとして知られている。開示される主題は、VVCの文脈で使用されてもよい。

【0055】

電子装置(320)および(330)は、他の構成要素(図示せず)を含むことができることを注意しておく。たとえば、電子装置(320)は、ビデオ・デコーダ(図示せず)を含むことができ、電子装置(330)は、ビデオ・エンコーダ(図示せず)も含むことができる。

【0056】

図4は、本開示のある実施形態によるビデオ・デコーダ(410)のブロック図を示す。ビデオ・デコーダ(410)は、電子装置(430)に含まれることができる。電子装置(430)は、受領器(431)(たとえば、受領回路)を含むことができる。ビデオ・デコーダ(410)は、図3の例におけるビデオ・デコーダ(310)の代わりに使用できる。

【0057】

受領器(431)は、ビデオ・デコーダ(410)によってデコードされるべき一つまたは複数の符号化ビデオ・シーケンスを受領してもよい;同じまたは別の実施形態において、一度に1つの符号化ビデオ・シーケンスであり、各符号化ビデオ・シーケンスのデコードは、他の符号化ビデオ・シーケンスから独立である。符号化ビデオ・シーケンスは、チャネル(401)から受信されてもよく、該チャネルは、エンコードされたビデオ・データを記憶する記憶装置へのハードウェア/ソフトウェア・リンクであってもよい。受領器(431)は、エンコードされたビデオ・データを、他のデータ、たとえば符号化されたオーディオ・データおよび/または補助データ・ストリームと一緒に受領してもよく、これらのデータは、それぞれの使用エンティティ(図示せず)を転送されてもよい。受領器(431)は、符号化ビデオ・シーケンスを他のデータから分離することができる。ネットワーク・ジッタ対策として、バッファメモリ(415)が、受領器(431)とエントローピー・デコーダ/パーサー(420)(以下「パーサー」)との間に結合されてもよい。ある種のアプリケーションでは、バッファメモリ(415)はビデオ・デコーダ(410)の一部である。他のアプリケーションでは、ビデオ・デコーダ(410)の外部にあることができる(図示せず)。さらに他のアプリケーションでは、たとえばネットワーク・ジッタに対抗するために、ビデオ・デコーダ(410)の外部にバッファメモリ(図示せず)があってもよく、さらに、たとえば再生タイミングを扱うために、ビデオ・デコーダ(410)の内部に別のバッファメモリ(415)があってもよい。受領器(431)が、十分な帯域幅および制御可能性の記憶/転送装置から、またはアイソクロナス・ネットワークからデータを受領している場合は、バッファメモリ(415)は、必要とされなくてもよく、または小さくてもよい。インターネットのようなベストエフォート型のパケット・ネットワークでの使用のためには、バッファメモリ(415)が要求されることがあり、比較的大きいことがあり、有利には適応サイズであることができ、少なくとも部分的に、ビデオ・デコーダ(410)の外部でオペレーティング・システムまたは同様の要素(図示せず)において実装されてもよい。

【0058】

ビデオ・デコーダ(410)は、符号化ビデオ・シーケンスからシンボル(421)を再構成するためのパーサー(420)を含んでいてもよい。これらのシンボルのカテゴリーは、ビデオ・デコーダ(410)の動作を管理するために使用される情報と、潜在的には、レンダリング装置(412)(たとえば表示画面)のようなレンダリング装置を制御するための情報

10

20

30

40

50

とを含む。レンダリング装置は、図4に示されたように、電子装置(430)の一体的な部分ではなく、電子装置(430)に結合されることができる。レンダリング装置(単数または複数)のための制御情報は、補足向上情報(Supplementary Enhancement Information)(SEIメッセージ)またはビデオユーザビリティ情報(Video Usability Information、VUI)パラメータ・セット・フラグメント(図示せず)の形であってもよい。パーサー(420)は、受領された符号化ビデオ・シーケンスをパースする/エントローピー復号することができる。符号化ビデオ・シーケンスの符号化は、ビデオ符号化技術または標準に従うことができ、可変長符号化、ハフマン符号化、コンテキスト感受性ありまたはなしの算術符号化などを含む、さまざまな原理に従うことができる。パーサー(420)は、符号化ビデオ・シーケンスから、ビデオ・デコーダ内のピクセルのサブグループのうちの少なくとも1つについてのサブグループ・パラメータのセットを、グループに対応する少なくとも1つのパラメータに基づいて、抽出することができる。サブグループは、ピクチャグループ(Group of Pictures、GOP)、ピクチャ、タイル、スライス、マクロブロック、符号化単位(Coding Unit、CU)、ブロック、変換単位(Transform Unit、TU)、予測単位(Prediction Unit、PU)などを含むことができる。パーサー(420)はまた、符号化ビデオ・シーケンスから、変換係数、量子化器パラメータ値、MV等の情報を抽出することができる。

【0059】

パーサー(420)は、バッファメモリ(415)から受領されたビデオ・シーケンスに対してエントローピー復号/パース動作を実行し、それによりシンボル(421)を生成することができる。

【0060】

シンボル(421)の再構成は、符号化されたビデオ・ピクチャまたはその諸部分のタイプ(たとえば、インターおよびイントラ・ピクチャ、インターおよびイントラ・ブロック)および他の要因に依存して、複数の異なるユニットに関わることができる。どのユニットがどのように関わるかは、符号化ビデオ・シーケンスからパーサー(420)によってパースされたサブグループ制御情報によって制御されることができる。パーサー(420)と下記の複数のユニットとの間のそのようなサブグループ制御情報の流れは、明確のため、描かれていない。

【0061】

すでに述べた機能ブロックのほかに、ビデオ・デコーダ(410)は、以下に説明するように、概念的に、いくつかの機能ユニットに分割できる。商業的制約の下で機能する実際的な実装では、これらのユニットの多くは互いに密接に相互作用し、少なくとも部分的に互いに統合されることができる。しかしながら、開示される主題を記述する目的のためには、下記の機能単位への概念的な細分が適切である。

【0062】

第1のユニットは、スケーラー/逆変換ユニット(451)である。スケーラー/逆変換ユニット(451)は、パーサー(420)から、量子化された変換係数および制御情報をシンボル(単数または複数)(421)として受領する。制御情報は、どの変換を使用するか、ブロック・サイズ、量子化因子、量子化スケーリング行列などを含む。スケーラー/逆変換ユニット(451)は、集計器(455)に入力できるサンプル値を含むブロックを出力することができる。

【0063】

場合によっては、スケーラー/逆変換(451)の出力サンプルは、イントラ符号化されたブロック、すなわち、以前に再構成されたピクチャからの予測情報を使用していないが、現在ピクチャの、以前に再構成された部分からの予測情報を使用することができるブロックに関与することができる。そのような予測情報は、イントラ・ピクチャ予測ユニット(452)によって提供されることができる。場合によっては、イントラ・ピクチャ予測ユニット(452)は、現在ピクチャ・バッファ(458)から取ってきた、周囲のすでに再構成された情報を使用して、再構成中のブロックと同じサイズおよび形状のブロッ

10

20

30

40

50

クを生成する。現在ピクチャー・バッファ(458)は、たとえば、部分的に再構成された現在ピクチャーおよび/または完全に再構成された現在ピクチャーをバッファリングする。集計器(455)は、場合によっては、サンプル毎に、イントラ予測ユニット(452)が生成した予測情報を、スケーラー/逆変換ユニット(451)によって提供される出力サンプル情報に加算する。

【0064】

他の場合には、スケーラー/逆変換ユニット(451)の出力サンプルは、インター符号化され、潜在的には動き補償されたブロックに關することができ。そのような場合、動き補償予測ユニット(453)は、予測のために使用されるサンプルを取ってくるために参照ピクチャー・メモリ(457)にアクセスすることができる。取ってきたサンプルを、ブロックに關するシンボル(421)に従って動き補償した後、これらのサンプルは、集計器(455)によってスケーラー/逆変換ユニットの出力(この場合、残差サンプルまたは残差信号と呼ばれる)に加算されて、それにより出力サンプル情報を生成することができる。動き補償ユニット(453)が予測サンプルを取ってくる参照ピクチャー・メモリ(457)内のアドレスは、シンボル(421)の形で動き補償ユニット(453)に利用可能なMVによって制御できる。該シンボルは、たとえばX、Y、および参照ピクチャー成分を有することができる。動き補償は、サンプル以下の正確なMVが使用されるとき参照ピクチャー・メモリ(457)から取ってこられるサンプル値の補間、MV予測機構などを含むことができる。

【0065】

集計器(455)の出力サンプルは、ループ・フィルタ・ユニット(456)内でさまざまなループ・フィルタリング技法にかけられることができる。ビデオ圧縮技術は、ループ内フィルタ技術を含むことができる。ループ内フィルタ技術は、符号化ビデオ・シーケンス(符号化されたビデオ・ビットストリームとも呼ばれる)に含まれるパラメータによって制御され、パーサー(420)からのシンボル(421)としてループ・フィルタ・ユニット(456)に利用可能にされるが、符号化されたピクチャーまたは符号化されたビデオ・シーケンスの(デコード順で)前の部分のデコード中に得られたメタ情報に応答するとともに、以前に再構成されループ・フィルタリングされたサンプル値に応答することもできる。

【0066】

ループ・フィルタ・ユニット(456)の出力はサンプル・ストリームであることができ、これは、レンダ装置(412)に出力されることができ、また将来のインターピクチャー予測において使用するために参照ピクチャー・メモリ(457)に記憶されることができ。

【0067】

符号化された画像は、いったん完全に再構成されると、将来の予測のための参照ピクチャーとして使用できる。たとえば、現在ピクチャーに対応する符号化されたピクチャーが完全に再構成され、該符号化されたピクチャーが(たとえば、パーサー(420)によって)参照ピクチャーとして同定されると、現在ピクチャー・バッファ(458)は参照ピクチャーメモリ(457)の一部となることができ、後続の符号化されたピクチャーの再構成を開始する前に、新鮮な現在ピクチャー・バッファが再割当てされることができ。

【0068】

ビデオ・デコーダ(410)は、ITU-T勧告H.265のような標準における所定のビデオ圧縮技術に従ってデコード動作を実行することができる。符号化ビデオ・シーケンスはビデオ圧縮技術または標準のシンタックスおよびビデオ圧縮技術または標準において文書化されているプロファイルに従うという意味で、符号化されたビデオ・シーケンスは、使用されているビデオ圧縮技術または標準によって規定されたシンタックスに準拠することができる。具体的には、プロファイルはビデオ圧縮技術または標準において利用可能なすべてのツールから、そのプロファイルのもとでの使用のためにそれだけが利用可能なツールとして、ある種のツールを選択することができる。準拠のためにはまた、符号化ビデオ・シーケンスの複雑さが、ビデオ圧縮技術または標準のレベルによって定義される範囲内にあ

10

20

30

40

50

ることも必要であることがある。いくつかの場合には、レベルは、最大ピクチャー・サイズ、最大フレーム・レート、最大再構成サンプル・レート（たとえば、毎秒メガサンプルの単位で測られる）、最大参照ピクチャー・サイズなどを制約する。レベルによって設定された限界は、場合によっては、符号化ビデオ・シーケンスにおいて信号伝達される、HRDバッファ管理のための仮設参照デコーダ（Hypothetical Reference Decoder、HRD）仕様およびメタデータを通じてさらに制約されることができる。

【0069】

ある実施形態において、受領器（431）は、エンコードされたビデオとともに追加の（冗長な）データを受領してもよい。追加データは、符号化されたビデオ・シーケンス（単数または複数）の一部として含まれていてもよい。追加データは、データを適正にデコードするため、および/またはもとのビデオ・データをより正確に再構成するために、ビデオ・デコーダ（410）によって使用されてもよい。追加データは、たとえば、時間的、空間的、または信号対雑音比（SNR）の向上層、冗長スライス、冗長ピクチャー、前方誤り訂正符号などの形でありうる。

【0070】

図5は、本開示のある実施形態によるビデオ・エンコーダ（503）のブロック図を示している。ビデオ・エンコーダ（503）は、電子装置（520）に含まれる。電子装置（520）は、送信器（540）（たとえば、送信回路）を含む。ビデオ・エンコーダ（503）は、図3の例におけるビデオ・エンコーダ（303）の代わりに使用できる。

【0071】

ビデオ・エンコーダ（503）は、ビデオ・エンコーダ（503）によって符号化されるべきビデオ画像を捕捉することができるビデオ源（501）（これは図5の例では電子装置（520）の一部ではない）からビデオ・サンプルを受領することができる。別の例では、ビデオ源（501）は、電子装置（520）の一部である。

【0072】

ビデオ源（501）は、任意の好適なビット深さ（たとえば、8ビット、10ビット、12ビット、...）、任意の色空間（たとえば、BT.601 YCrCb、RGB、...）および任意の好適なサンプリング構造（たとえば、YCrCb 4:2:0、YCrCb 4:4:4）でありうるデジタル・ビデオ・サンプル・ストリームの形で、ビデオ・エンコーダ（503）によって符号化されるべき源ビデオ・シーケンスを提供することができる。メディア・サービス・システムにおいては、ビデオ源（501）は、事前に準備されたビデオを記憶している記憶装置であってもよい。ビデオ会議システムにおいては、ビデオ源（501）は、ローカルでの画像情報をビデオ・シーケンスとして捕捉するカメラであってもよい。ビデオ・データは、シーケンスで見たときに動きを付与する複数の個々のピクチャーとして提供されてもよい。ピクチャー自体は、ピクセルの空間的アレイとして編成されてもよく、各ピクセルは、使用中のサンプリング構造、色空間などに依存して、一つまたは複数のサンプルを含むことができる。当業者は、ピクセルとサンプルとの間の関係を容易に理解することができる。下記の説明は、サンプルに焦点を当てる。

【0073】

ある実施形態によれば、ビデオ・エンコーダ（503）は、源ビデオ・シーケンスのピクチャーを、リアルタイムで、またはアプリケーションによって要求される任意の他の時間的制約の下で、符号化および圧縮して、符号化ビデオ・シーケンス（543）にすることができる。適切な符号化速度を施行することは、コントローラ（550）の一つの機能である。いくつかの実施形態では、コントローラ（550）は、以下に記載されるような他の機能ユニットを制御し、該他の機能ユニットに機能的に結合される。かかる結合は、明確のために描かれていない。コントローラ（550）によって設定されるパラメータは、レート制御に関連するパラメータ（ピクチャー・スキップ、量子化器、レート 歪み最適化技法のラムダ値、...）、ピクチャー・サイズ、ピクチャーグループ（GOP）レイアウト、最大MV許容参照領域などを含むことができる。コントローラ（550）は、ある種のシステム設計のために最適化されたビデオ・エンコーダ（503）に関する他の好適な機能を有するよ

10

20

30

40

50

うに構成できる。

【 0 0 7 4 】

いくつかの実施形態では、ビデオ・エンコーダ（503）は、符号化ループにおいて動作するように構成される。思い切って単純化した説明として、一例では、符号化ループは、源符号化器（530）（たとえば、符号化されるべき入力ピクチャーと参照ピクチャー（算数または複数）に基づいてシンボル・ストリームのようなシンボルを生成することを受け持つ）と、ビデオ・エンコーダ（503）に埋め込まれた（ローカル）デコーダ（533）とを含むことができる。デコーダ（533）は、（リモートの）デコーダも生成するであろうのと同様の仕方サンプル・データを生成するよう前記シンボルを再構成する（開示される主題において考慮されるビデオ圧縮技術では、シンボルと符号化ビデオ・ビットストリームとの間のどの圧縮も無損失である）。再構成されたサンプル・ストリーム（サンプル・データ）は、参照ピクチャー・メモリ（534）に入力される。シンボル・ストリームのデコードは、デコーダ位置（ローカルかリモートか）によらずビット正確な結果をもたらすので、参照ピクチャー・メモリ（534）の内容もローカル・エンコーダとリモート・エンコーダの間でビット正確である。言い換えると、エンコーダの予測部は、デコーダがデコード中に予測を使用するときに「見る」のとまったく同じサンプル値を参照ピクチャー・サンプルとして「見る」。参照ピクチャー同期性のこの基本原理（および、たとえば、チャンネルエラーのために同期性が維持できない場合の結果として生じるドリフト）は、いくつかの関連技術においても使用される。

10

【 0 0 7 5 】

「ローカル」デコーダ（533）の動作は、図4との関連ですでに上記で詳細に述べた「リモート」デコーダ、たとえばビデオ・デコーダ（410）の動作と同じであってよい。しかしながら、暫時図4も参照すると、シンボルが利用可能であり、エントロピー符号化器（545）およびパーサー（420）による、シンボルの符号化ビデオ・シーケンスへのエンコード/デコードが可逆でありうるので、バッファメモリ（415）およびパーサー（420）を含むビデオ・デコーダ（410）のエントロピー復号部は、ローカル・デコーダ（533）においては完全には実装されなくてもよい。

20

【 0 0 7 6 】

この時点で行なうことができる観察は、デコーダ内に存在するパース/エントロピー復号を除くどのデコーダ技術も、対応するエンコーダ内で実質的に同一の機能的形態で存在する必要があることである。このため、開示される主題はデコーダ動作に焦点を当てる。エンコーダ技術の記述は、包括的に記述されるデコーダ技術の逆であるため、短縮することができる。ある種の領域においてのみ、より詳細な説明が必要であり、以下に提供される。

30

【 0 0 7 7 】

動作中、いくつかの例では、源符号化器（530）は、「参照ピクチャー」として指定された、ビデオ・シーケンスからの一つまたは複数の以前に符号化されたピクチャーを参照して、入力ピクチャーを予測的に符号化する、動き補償された予測符号化を実行することができる。このようにして、符号化エンジン（532）は、入力ピクチャーのピクセル・ブロックと、入力ピクチャーに対する予測参照として選択されうる参照ピクチャー（単数または複数）のピクセル・ブロックとの間の差分を符号化する。

40

【 0 0 7 8 】

ローカル・ビデオ・デコーダ（533）は、源符号化器（530）によって生成されたシンボルに基づいて、参照ピクチャーとして指定されうるピクチャーの符号化されたビデオ・データをデコードすることができる。符号化エンジン（532）の動作は、有利には、損失のあるプロセスでありうる。符号化されたビデオ・データがビデオ・デコーダ（図5には示さず）でデコードされるとき、再構成されたビデオ・シーケンスは、典型的には、いくつかのエラーを伴う源ビデオ・シーケンスの複製でありうる。ローカル・ビデオ・デコーダ（533）は、ビデオ・デコーダによって参照ピクチャーに対して実行されうるデコード・プロセスを複製し、再構成された参照ピクチャーを参照ピクチャー・キャッシュ（53

50

4)に格納させることができる。このようにして、ビデオ・エンコーダ(503)は、遠端のビデオ・デコーダによって得られるであろう再構成された参照ピクチャーとしての共通の内容を(伝送エラーがなければ)有する再構成された参照ピクチャーのコピーを、ローカルに記憶することができる。

【0079】

予測器(535)は、符号化エンジン(532)について予測探索を実行することができる。すなわち、符号化されるべき新しいピクチャーについて、予測器(535)は、新しいピクチャーのための適切な予測参照のはたらきをしうるサンプル・データ(候補参照ピクセル・ブロックとして)またはある種のメタデータ、たとえば参照ピクチャーMV、ブロック形状などを求めて、参照ピクチャー・メモリ(534)を探索することができる。予測器(535)は、適切な予測参照を見出すために、サンプル・ブロック/ピクセル・ブロック毎に(on a sample block-by-pixel block basis)動作しうる。場合によっては、予測器(535)によって得られた検索結果によって決定されるところにより、入力ピクチャーは、参照ピクチャー・メモリ(534)に記憶された複数の参照ピクチャーから引き出された予測参照を有することができる。

【0080】

コントローラ(550)は、たとえば、ビデオ・データをエンコードするために使用されるパラメータおよびサブグループ・パラメータの設定を含め、源符号化器(530)の符号化動作を管理してもよい。

【0081】

上記の機能ユニットすべての出力は、エントロピー符号化器(545)におけるエントロピー符号化を受けることができる。エントロピー符号化器(545)は、ハフマン符号化、可変長符号化、算術符号化などといった技術に従ってシンボルを無損失圧縮することによって、さまざまな機能ユニットによって生成されたシンボルを符号化ビデオ・シーケンスに変換する。

【0082】

送信器(540)は、エントロピー符号化器(545)によって生成される符号化ビデオ・シーケンスをバッファに入れて、通信チャネル(560)を介した送信のために準備することができる。通信チャネル(560)は、エンコードされたビデオ・データを記憶する記憶装置へのハードウェア/ソフトウェア・リンクであってもよい。送信器(540)は、ビデオ符号化器(530)からの符号化されたビデオ・データを、送信されるべき他のデータ、たとえば符号化されたオーディオ・データおよび/または補助データ・ストリーム(源は図示せず)とマージすることができる。

【0083】

コントローラ(550)は、ビデオ・エンコーダ(503)の動作を管理してもよい。符号化の間、コントローラ(550)は、それぞれの符号化されたピクチャーに、ある符号化ピクチャー・タイプを割り当てることができる。符号化ピクチャー・タイプは、それぞれのピクチャーに適用されうる符号化技術に影響しうる。たとえば、ピクチャーはしばしば、以下のピクチャー・タイプのうちの1つとして割り当てられることがある。

【0084】

イントラピクチャー(Iピクチャー)は、予測の源としてシーケンス内の他のピクチャーを使用せずに、符号化され、デコードされうるものでありうる。いくつかのビデオ・コーデックは、たとえば、独立デコーダ・リフレッシュ(Independent Decoder Refresh、「IDR」)・ピクチャーを含む、異なるタイプのイントラ・ピクチャーを許容する。当業者は、Iピクチャーのこれらの変形、ならびにそれらのそれぞれの用途および特徴を認識する。

【0085】

予測ピクチャー(Pピクチャー)は、各ブロックのサンプル値を予測するために、最大で1つのMVおよび参照インデックスを用いるイントラ予測またはインター予測を用いて符号化およびデコードされうるものでありうる。

10

20

30

40

50

【 0 0 8 6 】

双方向予測ピクチャー（Bピクチャー）は、各ブロックのサンプル値を予測するために、最大で2つのMVおよび参照インデックスを用いるイントラ予測またはインター予測を用いて符号化およびデコードされうるものでありうる。同様に、マルチ予測ピクチャーは、単一のブロックの再構成のために、3つ以上の参照ピクチャーおよび関連するメタデータを使用することができる。

【 0 0 8 7 】

源ピクチャーは、普通、空間的に複数のサンプル・ブロック（たとえば、それぞれ4×4、8×8、4×8、または16×16サンプルのブロック）に分割され、ブロック毎に符号化されうる。ブロックは、ブロックのそれぞれのピクチャーに適用される符号化割り当てによって決定されるところにより、他の（すでに符号化された）ブロックを参照して予測的に符号化されうる。たとえば、Iピクチャーのブロックは、非予測的に符号化されてもよく、または、同じピクチャーのすでに符号化されたブロックを参照して予測的に符号化されてもよい（空間的予測またはイントラ予測）。Pピクチャーのピクセル・ブロックは、以前に符号化された一つの参照ピクチャーを参照して、空間的予測を介してまたは時間的予測を介して予測的に符号化されてもよい。Bピクチャーのブロックは、1つまたは2つの以前に符号化された参照ピクチャーを参照して、空間的予測を介して、または時間的予測を介して予測的に符号化されてもよい。

【 0 0 8 8 】

ビデオ・エンコーダ（503）は、ITU-T勧告H.265などの所定のビデオ符号化技術または標準に従って符号化動作を実行することができる。その動作において、ビデオ・エンコーダ（503）は、入力ビデオ・シーケンスにおける時間的および空間的冗長性を活用する予測符号化動作を含む、さまざまな圧縮動作を実行することができる。よって、符号化されたビデオ・データは、使用されるビデオ符号化技術または標準によって指定されるシンタックスに準拠しうる。

【 0 0 8 9 】

ある実施形態において、送信器（540）は、エンコードされたビデオと一緒に追加データを送信してもよい。源符号化器（530）は、符号化ビデオ・シーケンスの一部としてそのようなデータを含めてもよい。追加データは、時間的／空間的／SNR向上層、冗長ピクチャーおよびスライスのような他の形の冗長データ、SEIメッセージ、VUIパラメータ・セット・フラグメントなどを含んでいてもよい。

【 0 0 9 0 】

ビデオは、時間的シーケンスにおいて複数の源ピクチャー（ビデオ・ピクチャー）として捕捉されてもよい。ピクチャー内予測（しばしば、イントラ予測と略される）は、所与のピクチャーにおける空間的相関を利用し、ピクチャー間予測は、ピクチャー間の（時間的または他の）相関を利用する。一例では、現在ピクチャーと呼ばれるエンコード／デコード対象の特定のピクチャーは、ブロックに分割される。現在ピクチャー内のブロックが、ビデオにおける、前に符号化され、かつ、まだバッファに入れられている参照ピクチャー内の参照ブロックに類似する場合、現在ピクチャー内のそのブロックは、MVと呼ばれるベクトルによって符号化できる。MVは、参照ピクチャー内の参照ブロックをポイントし、複数の参照ピクチャーが使用される場合には、参照ピクチャーを同定する第3の次元を有することができる。

【 0 0 9 1 】

いくつかの実施形態において、ピクチャー間予測において双予測技術が使用できる。双予測技術によれば、いずれもビデオにおいて現在ピクチャーよりデコード順で先行する（ただし、表示順では、それぞれ過去および将来であってもよい）第1の参照ピクチャーおよび第2の参照ピクチャーのような2つの参照ピクチャーが使用される。現在ピクチャー内のブロックは、第1の参照ピクチャー内の第1の参照ブロックをポイントする第1 MVと、第2の参照ピクチャー内の第2の参照ブロックをポイントする第2 MVとによって符号化できる。ブロックは、第1の参照ブロックと第2の参照ブロックの組み合わせによって予測で

10

20

30

40

50

きる。

【0092】

さらに、符号化効率を改善するために、ピクチャー間予測においてマージモード技法が使用できる。

【0093】

本開示のいくつかの実施形態によれば、ピクチャー間予測およびピクチャー内予測などの予測は、ブロックの単位で実行される。たとえば、HEVC規格によれば、ビデオ・ピクチャーのシーケンスにおけるピクチャーは、圧縮のために符号化ツリー単位（CTU）に分割され、ピクチャーにおけるそれらのCTUは、 64×64 ピクセル、 32×32 ピクセル、または 16×16 ピクセルなどの同じサイズを有する。一般に、CTUは、1つのルーマCTBおよび2つのクロマCTBである3つの符号化ツリーブロック（CTB）を含む。各CTUは、再帰的に、一つまたは複数の符号化単位（CU）に四分木分割されていくことができる。たとえば、 64×64 ピクセルのCTUは、 64×64 ピクセルの1つのCU、または 32×32 ピクセルの4つのCU、または 16×16 ピクセルの16個のCUに分割されることができる。一例では、各CUは、インター予測タイプまたはイントラ予測タイプのような、そのCUについての予測タイプを決定するために解析される。CUは時間的および/または空間的予測可能性に依存して、一つまたは複数の予測単位（PU）に分割される。一般に、各PUはルーマ予測ブロック（PB）および2つのクロマPBを含む。ある実施形態では、符号化（エンコード/デコード）における予測動作は、予測ブロックの単位で実行される。予測ブロックの例としてルーマ予測ブロックを用いると、予測ブロックは、 8×8 ピクセル、 16×16 ピクセル、 8×16 ピクセル、 16×8 ピクセルなど、ピクセルについての値（たとえば、ルーマ値）の行列を含む。

【0094】

図6は、本開示の別の実施形態によるビデオ・エンコーダ（603）の図を示す。ビデオ・エンコーダ（603）は、ビデオ・ピクチャーのシーケンス内の現在ビデオ・ピクチャー内のサンプル値の処理ブロック（たとえば、予測ブロック）を受領し、処理ブロックを、符号化ビデオ・シーケンスの一部である符号化されたピクチャーにエンコードするように構成される。一例では、ビデオ・エンコーダ（603）は、図3の例におけるビデオ・エンコーダ（303）の代わりに使用される。

【0095】

HEVCの例では、ビデオ・エンコーダ（603）は、 8×8 サンプルなどの予測ブロックのような処理ブロックについてサンプル値の行列を受領する。ビデオ・エンコーダ（603）は、処理ブロックが、イントラモード、インターモード、または双予測モードのどれを使用して、最もよく符号化されるかを、たとえばレート 歪み最適化を使用して、判別する。処理ブロックがイントラモードで符号化される場合、ビデオ・エンコーダ（603）は、処理ブロックを符号化されたピクチャーにエンコードするためにイントラ予測技法を使用してもよく；処理ブロックがインターモードまたは双予測モードで符号化される場合、ビデオ・エンコーダ（603）は、処理ブロックを符号化されたピクチャーにエンコードするために、それぞれ、インター予測技法または双予測技法を使用してもよい。ある種のビデオ符号化技術では、マージモード（merge mode）は、MVが一つまたは複数のMV予測子から導出されるが前記予測子の外の符号化されたMV成分の利益のない、ピクチャー間予測サブモードでありうる。ある種の他のビデオ符号化技術では、対象ブロックに適用可能なMV成分が存在してもよい。一例では、ビデオ・エンコーダ（603）は、処理ブロックのモードを決定するためのモード決定モジュール（図示せず）などの他のコンポーネントを含む。

【0096】

図6の例では、ビデオ・エンコーダ（603）は、インター・エンコーダ（630）、イントラ・エンコーダ（622）、残差計算器（623）、スイッチ（626）、残差エンコーダ（624）、全般コントローラ（621）、およびエントロピー符号化器（625）を、図6に示されるように一緒に結合されて含む。

10

20

30

40

50

【 0 0 9 7 】

インター・エンコーダ(630)は、現在ブロック(たとえば、処理ブロック)のサンプルを受領し、該ブロックを参照ピクチャー内の一つまたは複数の参照ブロック(たとえば、以前のピクチャーおよび後のピクチャー内のブロック)と比較し、インター予測情報(たとえば、インター・エンコード技術による冗長情報の記述、MV、マージモード情報)を生成し、該インター予測情報に基づいて、任意の好適な技法を使用してインター予測結果(たとえば、予測ブロック)を計算するように構成される。いくつかの例では、前記参照ピクチャーは、エンコードされたビデオ情報に基づいてデコードされた、デコードされた参照ピクチャーである。

【 0 0 9 8 】

イントラ・エンコーダ(622)は、現在ブロック(たとえば、処理ブロック)のサンプルを受領し、場合によっては、該ブロックを、同じピクチャー内ですでに符号化されているブロックと比較し、変換後に量子化された係数を生成し、場合によっては、イントラ予測情報(たとえば、一つまたは複数のイントラ・エンコード技法によるイントラ予測方向情報)も生成するように構成される。一例では、イントラ・エンコーダ(622)はまた、該イントラ予測情報および同じピクチャー内の参照ブロックに基づいて、イントラ予測結果(たとえば、予測ブロック)を計算する。

【 0 0 9 9 】

全般コントローラ(621)は、全般制御データを決定し、全般制御データに基づいてビデオ・エンコーダ(603)の他のコンポーネントを制御するように構成される。一例では、全般コントローラ(621)は、ブロックのモードを決定し、そのモードに基づいて制御信号をスイッチ(626)に提供する。たとえば、モードがイントラモードである場合、全般コントローラ(621)は、残差計算器(623)による使用のためにイントラモードの結果を選択するようスイッチ(626)を制御し、イントラ予測情報を選択し、イントラ予測情報をビットストリームに含めるようエン트로ピー・エンコーダ(625)を制御する；モードがインターモードである場合、全般コントローラ(621)は、残差計算器(623)による使用のためにインター予測の結果を選択するようスイッチ(626)を制御し、インター予測情報を選択し、インター予測情報をビットストリームに含めるようエン트로ピー・エンコーダ(625)を制御する。

【 0 1 0 0 】

残差計算器(623)は、受領されたブロックと、イントラ・エンコーダ(622)またはインター・エンコーダ(630)から選択された予測結果との差(残差データ)を計算するように構成される。残差エンコーダ(624)は、残差データに基づいて、残差データをエンコードして変換係数を生成するように構成される。一例では、残差エンコーダ(624)は、残差データを空間領域から周波数領域に変換し、変換係数を生成するように構成される。次いで、変換係数は、量子化処理にかけられ、量子化された変換係数を得る。さまざまな実施形態において、ビデオ・エンコーダ(603)は、残差デコーダ(628)をも含む。残差デコーダ(628)は、逆変換を実行して、デコードされた残差データを生成するように構成される。デコードされた残差データは、イントラ・エンコーダ(622)およびインター・エンコーダ(630)によって好適に使用されることができる。たとえば、インター・エンコーダ(630)は、デコードされた残差データおよびインター予測情報に基づいて、デコードされたブロックを生成することができ、イントラ・エンコーダ(622)は、デコードされた残差データおよびイントラ予測情報に基づいて、デコードされたブロックを生成することができ、デコードされたブロックは、デコードされたピクチャーを生成するために好適に処理され、デコードされたピクチャーは、メモリ回路(図示せず)内でバッファリングされ、いくつかの例では参照ピクチャーとして使用されることができる。

【 0 1 0 1 】

エン트로ピー・エンコーダ(625)は、エンコードされたブロックを含むようにビットストリームをフォーマットするように構成される。エン트로ピー・エンコーダ(625)は、HEVCのような好適な標準に従ってさまざまな情報を含めるように構成される。一例で

10

20

30

40

50

は、エントロピー・エンコーダ(625)は、全般制御データ、選択された予測情報(たとえば、イントラ予測情報またはインター予測情報)、残差情報、および他の好適な情報をビットストリーム内に含めるように構成される。開示される主題によれば、インターモードまたは双予測モードのいずれかのマージ・サブモードにおいてブロックを符号化する場合、残差情報は存在しないことを注意しておく。

【0102】

図7は、本開示の別の実施形態によるビデオ・デコーダ(710)の図を示す。ビデオ・デコーダ(710)は、符号化されたビデオ・シーケンスの一部である符号化されたピクチャーを受領し、符号化されたピクチャーをデコードして、再構成されたピクチャーを生成するように構成される。一例では、ビデオ・デコーダ(710)は、図3の例におけるビデオ・デコーダ(310)の代わりに使用される。

10

【0103】

図7の例では、ビデオ・デコーダ(710)は、エントロピー・デコーダ(771)、インター・デコーダ(780)、残差デコーダ(773)、再構成モジュール(774)、およびイントラ・デコーダ(772)が図7に示されるように一緒に結合されたものを含む。

【0104】

エントロピー・デコーダ(771)は、符号化されたピクチャーから、その符号化されたピクチャーが構成されるシンタックス要素を表わすある種のシンボルを再構成するように構成されることができる。そのようなシンボルは、たとえば、ブロックが符号化されるモード(たとえば、イントラモード、インターモード、双予測モード、マージ・サブモードまたは別のサブモードにおける後者の2つ)、イントラ・デコーダ(772)またはインター・デコーダ(780)によってそれぞれ予測のために使用されるある種のサンプルまたはメタデータを同定することができる予測情報(たとえば、イントラ予測情報またはインター予測情報など)、たとえば量子化された変換係数の形の残差情報などを含むことができる。一例では、予測モードがインターまたは双予測モードである場合、インター予測情報がインター・デコーダ(780)に提供され;予測タイプがイントラ予測タイプである場合には、イントラ予測情報がイントラ・デコーダ(772)に提供される。残差情報は、逆量子化を受けることができ、残差デコーダ(773)に提供される。

20

【0105】

インター・デコーダ(780)は、インター予測情報を受領し、該インター予測情報に基づいてインター予測結果を生成するように構成される。

30

【0106】

イントラ・デコーダ(772)は、イントラ予測情報を受領し、該イントラ予測情報に基づいて予測結果を生成するように構成される。

【0107】

残差デコーダ(773)は、逆量子化を実行して量子化解除された変換係数を抽出し、量子化解除された変換係数を処理して、残差を周波数領域から空間領域に変換するように構成される。残差デコーダ(773)はまた、ある種の制御情報(量子化器パラメータ(QP)を含む)をも必要とすることがあり、その情報は、エントロピー・デコーダ(771)によって提供されてもよい(これは、低ボリュウムの制御情報のみであるため、データ経路は描かれていない)。

40

【0108】

再構成モジュール(774)は、空間領域において、残差デコーダ(773)によって出力される残差と、予測結果(場合に応じてイントラまたはインター予測モジュールによって出力される)とを組み合わせ、再構成されたブロックを形成するように構成され、該再構成されたブロックは再構成されたピクチャーの一部であってもよく、該再構成されたピクチャーは再構成されたビデオの一部であってもよい。視覚的品質を改善するためにブロック解除動作などの他の好適な動作が実行されることができることを注意しておく。

【0109】

なお、ビデオ・エンコーダ(303)、(503)、(603)、およびビデオ・デコーダ(

50

310)、(410)、(710)は、任意の好適な技法を用いて実装できる。ある実施形態では、ビデオ・エンコーダ(303)、(503)、(603)およびビデオ・デコーダ(310)、(410)、(710)は、一つまたは複数の集積回路を使用して実装できる。別の実施形態では、ビデオ・エンコーダ(303)、(503)、(603)、およびビデオ・デコーダ(310)、(410)、(710)は、ソフトウェア命令を実行する一つまたは複数のプロセッサを使用して実装できる。

【0110】

II. 例示的な高レベル構文要素

【0111】

表1は、VVCのようないくつかの関連する例における例示的なシーケンスパラメータセット(SPS)構文要素を示す。イントラ符号化およびインター符号化に関連する構文要素の両方が表1に含まれる。イントラスライスのみを含むイントラ・プロファイルについては、インター符号化構文要素がSPSに存在することがあるが、イントラ・プロファイルのデコード・プロセスにおいて使用されないことに留意されたい。この状況は、任意の静止ピクチャーのプロファイルにも適用される。すなわち、イントラスライス(単数または複数)のみを含む静止ピクチャー・プロファイルについて、インター符号化構文要素は、静止プロファイルのデコード・プロセスにおいて使用されない。

表1 シーケンスパラメータセットの生のバイト・シーケンス・ペイロード(raw byte sequence payload、RBSP)構文

【表1】

seq_parameter_set_rbsp() {	記述子
sps_seq_parameter_set_id	u(4)
sps_video_parameter_set_id	u(4)
sps_max_sublayers_minus1	u(3)
sps_reserved_zero_4bits	u(4)
sps_ptl_dpb_hrd_params_present_flag	u(1)
if(sps_ptl_dpb_hrd_params_present_flag)	
profile_tier_level(1, sps_max_sublayers_minus1)	
gdr_enabled_flag	u(1)
chroma_format_idc	u(2)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
res_change_in_clvs_allowed_flag	u(1)
pic_width_max_in_luma_samples	ue(v)
pic_height_max_in_luma_samples	ue(v)
sps_conformance_window_flag	u(1)
if(sps_conformance_window_flag) {	
sps_conf_win_left_offset	ue(v)
sps_conf_win_right_offset	ue(v)
sps_conf_win_top_offset	ue(v)
sps_conf_win_bottom_offset	ue(v)
}	
sps_log2_ctu_size_minus5	u(2)
subpic_info_present_flag	u(1)
if(subpic_info_present_flag) {	
sps_num_subpics_minus1	ue(v)
sps_independent_subpics_flag	u(1)
for(i = 0; sps_num_subpics_minus1 > 0 && i <= sps_num_subpics_minus1; i++) {	
if(i > 0 && pic_width_max_in_luma_samples > CtbSizeY)	
subpic_ctu_top_left_x[i]	u(v)
if(i > 0 && pic_height_max_in_luma_samples > CtbSizeY) {	
subpic_ctu_top_left_y[i]	u(v)
if(i < sps_num_subpics_minus1 && pic_width_max_in_luma_samples > CtbSizeY)	
subpic_width_minus1[i]	u(v)
if(i < sps_num_subpics_minus1 && pic_height_max_in_luma_samples > CtbSizeY)	
subpic_height_minus1[i]	u(v)
if(!sps_independent_subpics_flag) {	

10

20

30

40

50

subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}	
sps_subpic_id_len_minus1	ue(v)
subpic_id_mapping_explicitly_signalled_flag	u(1)
if(subpic_id_mapping_explicitly_signalled_flag) {	
subpic_id_mapping_in_sps_flag	u(1)
if(subpic_id_mapping_in_sps_flag)	
for(i = 0; i <= sps_num_subpics_minus1; i++)	
sps_subpic_id[i]	u(v)
}	
}	
bit_depth_minus8	ue(v)
sps_entropy_coding_sync_enabled_flag	u(1)
if(sps_entropy_coding_sync_enabled_flag)	
sps_wpp_entry_point_offsets_present_flag	u(1)
sps_weighted_pred_flag	u(1)
sps_weighted_bipred_flag	u(1)
log2_max_pic_order_cnt_lsb_minus4	u(4)
sps_poc_msb_flag	u(1)
if(sps_poc_msb_flag)	
poc_msb_len_minus1	ue(v)
num_extra_ph_bits_bytes	u(2)
extra_ph_bits_struct(num_extra_ph_bits_bytes)	
num_extra_sh_bits_bytes	u(2)
extra_sh_bits_struct(num_extra_sh_bits_bytes)	
if(sps_max_sublayers_minus1 > 0)	
sps_sublayer_dpb_params_flag	u(1)
if(sps_ptl_dpb_hrd_params_present_flag)	
dpb_parameters(sps_max_sublayers_minus1, sps_sublayer_dpb_params_flag)	
long_term_ref_pics_flag	u(1)
inter_layer_ref_pics_present_flag	u(1)
sps_idr_rpl_present_flag	u(1)
rpl1_same_as_rpl0_flag	u(1)
for(i = 0; i < rpl1_same_as_rpl0_flag ? 1 : 2; i++) {	
num_ref_pic_lists_in_sps[i]	ue(v)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
ref_pic_list_struct(i, j)	
}	
if(ChromaArrayType != 0)	
qtbtt_dual_tree_intra_flag	u(1)

10

20

30

40

50

log2_min_luma_coding_block_size_minus2	ue(v)
partition_constraints_override_enabled_flag	u(1)
sps_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
sps_log2_diff_min_qt_min_cb_inter_slice	ue(v)
sps_max_mtt_hierarchy_depth_inter_slice	ue(v)
if(sps_max_mtt_hierarchy_depth_inter_slice != 0) {	
sps_log2_diff_max_bt_min_qt_inter_slice	ue(v)
sps_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
if(qtbt_dual_tree_intra_flag) {	
sps_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
sps_max_luma_transform_size_64_flag	u(1)
if(ChromaArrayType != 0) {	
sps_joint_cbr_enabled_flag	u(1)
same_qp_table_for_chroma	u(1)
numQpTables = same_qp_table_for_chroma ? 1 : (sps_joint_cbr_enabled_flag ? 3 : 2)	
for(i = 0; i < numQpTables; i++) {	
qp_table_start_minus26[i]	se(v)
num_points_in_qp_table_minus1[i]	ue(v)
for(j = 0; j <= num_points_in_qp_table_minus1[i]; j++) {	
delta_qp_in_val_minus1[i][j]	ue(v)
delta_qp_diff_val[i][j]	ue(v)
}	
}	
}	
sps_sao_enabled_flag	u(1)
sps_alf_enabled_flag	u(1)
if(sps_alf_enabled_flag && ChromaArrayType != 0)	
sps_ccalf_enabled_flag	u(1)
sps_transform_skip_enabled_flag	u(1)
if(sps_transform_skip_enabled_flag) {	

10

20

30

40

50

log2_transform_skip_max_size_minus2	ue(v)
sps_bdpcm_enabled_flag	u(1)
}	
sps_ref_wraparound_enabled_flag	u(1)
sps_temporal_mvp_enabled_flag	u(1)
if(sps_temporal_mvp_enabled_flag)	
sps_sbtmvp_enabled_flag	u(1)
sps_amvr_enabled_flag	u(1)
sps_bdof_enabled_flag	u(1)
if(sps_bdof_enabled_flag)	
sps_bdof_pic_present_flag	u(1)
sps_smvd_enabled_flag	u(1)
sps_dmvr_enabled_flag	u(1)
if(sps_dmvr_enabled_flag)	
sps_dmvr_pic_present_flag	u(1)
sps_mmvd_enabled_flag	u(1)
sps_isp_enabled_flag	u(1)
sps_mrl_enabled_flag	u(1)
sps_mip_enabled_flag	u(1)
if(ChromaArrayType != 0)	
sps_cclm_enabled_flag	u(1)
if(chroma_format_idc == 1) {	
sps_chroma_horizontal_collocated_flag	u(1)
sps_chroma_vertical_collocated_flag	u(1)
}	
sps_mts_enabled_flag	u(1)
if(sps_mts_enabled_flag) {	
sps_explicit_mts_intra_enabled_flag	u(1)
sps_explicit_mts_inter_enabled_flag	u(1)
}	
six_minus_max_num_merge_cand	ue(v)
sps_sbt_enabled_flag	u(1)
sps_affine_enabled_flag	u(1)
if(sps_affine_enabled_flag) {	
five_minus_max_num_subblock_merge_cand	ue(v)
sps_affine_type_flag	u(1)
if(sps_amvr_enabled_flag)	
sps_affine_amvr_enabled_flag	u(1)
sps_affine_prof_enabled_flag	u(1)
if(sps_affine_prof_enabled_flag)	
sps_prof_pic_present_flag	u(1)
}	

10

20

30

40

50

sps_palette_enabled_flag	u(1)
if(ChromaArrayType == 3 && !sps_max_luma_transform_size_64_flag)	
sps_act_enabled_flag	u(1)
if(sps_transform_skip_enabled_flag sps_palette_enabled_flag)	
min_qp_prime_ts_minus4	ue(v)
sps_bcw_enabled_flag	u(1)
sps_ibc_enabled_flag	u(1)
if(sps_ibc_enabled_flag)	
six_minus_max_num_ibc_merge_cand	ue(v)
sps_ciip_enabled_flag	u(1)
if(sps_mmvd_enabled_flag)	
sps_fpel_mmvd_enabled_flag	u(1)
if(MaxNumMergeCand >= 2) {	
sps_gpm_enabled_flag	u(1)
if(sps_gpm_enabled_flag && MaxNumMergeCand >= 3)	
max_num_merge_cand_minus_max_num_gpm_cand	ue(v)
}	
sps_lmcs_enabled_flag	u(1)
sps_lfst_enabled_flag	u(1)
sps_ladf_enabled_flag	u(1)
if(sps_ladf_enabled_flag) {	
sps_num_ladf_intervals_minus2	u(2)
sps_ladf_lowest_interval_qp_offset	se(v)
for(i = 0; i < sps_num_ladf_intervals_minus2 + 1; i++) {	
sps_ladf_qp_offset[i]	se(v)
sps_ladf_delta_threshold_minus1[i]	ue(v)
}	
}	
log2_parallel_merge_level_minus2	ue(v)
sps_explicit_scaling_list_enabled_flag	u(1)
sps_dep_quant_enabled_flag	u(1)
if(!sps_dep_quant_enabled_flag)	
sps_sign_data_hiding_enabled_flag	u(1)
sps_virtual_boundaries_enabled_flag	u(1)
if(sps_virtual_boundaries_enabled_flag) {	
sps_virtual_boundaries_present_flag	u(1)
if(sps_virtual_boundaries_present_flag) {	
sps_num_ver_virtual_boundaries	u(2)
for(i = 0; i < sps_num_ver_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_x[i]	u(13)
sps_num_hor_virtual_boundaries	u(2)
for(i = 0; i < sps_num_hor_virtual_boundaries; i++)	

10

20

sps_virtual_boundaries_pos_y[i]	u(13)
}	
}	
if(sps_ptl_dpb_hrd_params_present_flag) {	
sps_general_hrd_params_present_flag	u(1)
if(sps_general_hrd_params_present_flag) {	
general_hrd_parameters()	
if(sps_max_sublayers_minus1 > 0)	
sps_sublayer_cpb_params_present_flag	u(1)
firstSubLayer = sps_sublayer_cpb_params_present_flag ? 0 :	
sps_max_sublayers_minus1	
ols_hrd_parameters(firstSubLayer, sps_max_sublayers_minus1)	
}	
}	
field_seq_flag	u(1)
vui_parameters_present_flag	u(1)
if(vui_parameters_present_flag)	
vui_parameters() /* Specified in ITU-T H.266 ISO/IEC 23002-7 */	
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

30

40

【 0 1 1 2 】

表2は、VVCのようないくつかの関連した例における例示的なピクチャーパラメータセット（PPS）構文要素を示す。イントラ符号化およびインター符号化に関連する構文要素の両方が表2に含まれている。イントラスライスのみを含むイントラ・プロファイルについては、インター符号化構文要素がPPSに存在することがあるが、イントラ・プロファイルのデコード・プロセスにおいて使用されないことに留意されたい。この状況は、任意の静止ピクチャーのプロファイルにも適用される。すなわち、イントラスライス（単数または複数）のみを含む静止ピクチャー・プロファイルについて、インター符号化構文要素は

50

、静止プロファイルのデコード・プロセスにおいて使用されない。

表2 ピクチャー・パラメータRBSP構文

【表2】

pic_parameter_set_rbsp() {	記述子
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	u(4)
mixed_nalu_types_in_pic_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
pps_conformance_window_flag	u(1)
if(pps_conformance_window_flag) {	
pps_conf_win_left_offset	ue(v)
pps_conf_win_right_offset	ue(v)
pps_conf_win_top_offset	ue(v)
pps_conf_win_bottom_offset	ue(v)
}	
scaling_window_explicit_signalling_flag	u(1)
if(scaling_window_explicit_signalling_flag) {	
scaling_win_left_offset	ue(v)
scaling_win_right_offset	ue(v)
scaling_win_top_offset	ue(v)
scaling_win_bottom_offset	ue(v)
}	
output_flag_present_flag	u(1)
subpic_id_mapping_in_pps_flag	u(1)
if(subpic_id_mapping_in_pps_flag) {	
pps_num_subpics_minus1	ue(v)
pps_subpic_id_len_minus1	ue(v)
for(i = 0; i <= pps_num_subpic_minus1; i++)	
pps_subpic_id[i]	u(v)
}	
no_pic_partition_flag	u(1)
if(!no_pic_partition_flag) {	
pps_log2_ctu_size_minus5	u(2)
num_exp_tile_columns_minus1	ue(v)
num_exp_tile_rows_minus1	ue(v)
for(i = 0; i <= num_exp_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for(i = 0; i <= num_exp_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
if(NumTilesInPic > 1)	

10

20

rect_slice_flag	u(1)
if(rect_slice_flag)	
single_slice_per_subpic_flag	u(1)
if(rect_slice_flag && !single_slice_per_subpic_flag) {	
num_slices_in_pic_minus1	ue(v)
if(num_slices_in_pic_minus1 > 0)	
tile_idx_delta_present_flag	u(1)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
if(NumTileColumns > 1)	
slice_width_in_tiles_minus1[i]	ue(v)
if(NumTileRows > 1 && (tile_idx_delta_present_flag SliceTopLeftTileIdx[i] % NumTileColumns == 0))	
slice_height_in_tiles_minus1[i]	ue(v)
if(slice_width_in_tiles_minus1[i] == 0 && slice_height_in_tiles_minus1[i] == 0 && RowHeight[SliceTopLeftTileIdx[i] / NumTileColumns] > 1) {	
num_exp_slices_in_tile[i]	ue(v)
for(j = 0; j < num_exp_slices_in_tile[i]; j++)	
exp_slice_height_in_ctus_minus1[i][j]	ue(v)
i += NumSlicesInTile[i] - 1	
}	
if(tile_idx_delta_present_flag && i < num_slices_in_pic_minus1)	
tile_idx_delta[i]	se(v)
}	
}	
loop_filter_across_tiles_enabled_flag	u(1)
loop_filter_across_slices_enabled_flag	u(1)
}	
cabac_init_present_flag	u(1)
for(i = 0; i < 2; i++)	
num_ref_idx_default_active_minus1[i]	ue(v)
rpl1_idx_present_flag	u(1)
init_qp_minus26	se(v)
cu_qp_delta_enabled_flag	u(1)
pps_chroma_tool_offsets_present_flag	u(1)
if(pps_chroma_tool_offsets_present_flag) {	
pps_cb_qp_offset	se(v)
pps_cr_qp_offset	se(v)
pps_joint_cbr_qp_offset_present_flag	u(1)
if(pps_joint_cbr_qp_offset_present_flag)	
pps_joint_cbr_qp_offset_value	se(v)
pps_slice_chroma_qp_offsets_present_flag	u(1)

30

40

50

pps_cu_chroma_qp_offset_list_enabled_flag	u(1)
}	
if(pps_cu_chroma_qp_offset_list_enabled_flag) {	
chroma_qp_offset_list_len_minus1	ue(v)
for(i = 0; i <= chroma_qp_offset_list_len_minus1; i++) {	
cb_qp_offset_list[i]	se(v)
cr_qp_offset_list[i]	se(v)
if(pps_joint_cbr_qp_offset_present_flag)	
joint_cbr_qp_offset_list[i]	se(v)
}	
}	
pps_weighted_pred_flag	u(1)
pps_weighted_bipred_flag	u(1)
deblocking_filter_control_present_flag	u(1)
if(deblocking_filter_control_present_flag) {	
deblocking_filter_override_enabled_flag	u(1)
pps_deblocking_filter_disabled_flag	u(1)
if(!pps_deblocking_filter_disabled_flag) {	
pps_beta_offset_div2	se(v)
pps_tc_offset_div2	se(v)
pps_cb_beta_offset_div2	se(v)
pps_cb_tc_offset_div2	se(v)
pps_cr_beta_offset_div2	se(v)
pps_cr_tc_offset_div2	se(v)
}	
}	
rpl_info_in_ph_flag	u(1)
if(deblocking_filter_override_enabled_flag)	
dbf_info_in_ph_flag	u(1)
sao_info_in_ph_flag	u(1)
alf_info_in_ph_flag	u(1)
if((pps_weighted_pred_flag pps_weighted_bipred_flag) && rpl_info_in_ph_flag)	
wp_info_in_ph_flag	u(1)
qp_delta_info_in_ph_flag	u(1)
pps_ref_wraparound_enabled_flag	u(1)
if(pps_ref_wraparound_enabled_flag)	
pps_ref_wraparound_offset	ue(v)
picture_header_extension_present_flag	u(1)
slice_header_extension_present_flag	u(1)
pps_extension_flag	u(1)
if(pps_extension_flag)	
while(more_rbsp_data())	

10

20

pps_extension_data_flag	u(1)
rbp_trailing_bits()	
}	

30

【 0 1 1 3 】

表3および表4は、VVCのようないくつかの関連する例における例示的なピクチャーヘッダ（PH）構文要素を示す。ピクチャーは、イントラスライスとインタースライスの組み合わせを含むことがある。ピクチャーがイントラスライスのみを含む場合、インター符号化構文要素がPHに存在することがあるが、ピクチャーのデコード・プロセスにおいて使用されない。ピクチャーにおけるオーバーヘッドを減らすために、いくつかの例では、イントラ符号化に関連する構文要素とインター符号化に関連する構文要素を条件付きで信号伝達するために、表4におけるph_inter_slice_allowed_flagとph_intra_slice_allowed_flagが使用される。

表3 ピクチャーヘッダRBSP構文

40

【 表 3 】

picture_header_rbsp() {	記述子
picture_header_structure()	
rbp_trailing_bits()	
}	

表4 ピクチャーヘッダ構造

50

【表 4】

picture_header_structure() {	記述子
gdr_or_irap_pic_flag	u(1)
if(gdr_or_irap_pic_flag)	
gdr_pic_flag	u(1)
ph_inter_slice_allowed_flag	u(1)
if(ph_inter_slice_allowed_flag)	
ph_intra_slice_allowed_flag	u(1)
non_reference_picture_flag	u(1)
ph_pic_parameter_set_id	ue(v)
ph_pic_order_cnt_lsb	u(v)
if(gdr_or_irap_pic_flag)	
no_output_of_prior_pics_flag	u(1)
if(gdr_pic_flag)	
recovery_poc_cnt	ue(v)
for(i = 0; i < NumExtraPhBits; i++)	
ph_extra_bit[i]	u(1)
if(sps_poc_msb_flag) {	
ph_poc_msb_present_flag	u(1)
if(ph_poc_msb_present_flag)	
poc_msb_val	u(v)
}	
if(sps_alf_enabled_flag && alf_info_in_ph_flag) {	
ph_alf_enabled_flag	u(1)
if(ph_alf_enabled_flag) {	
ph_num_alf_ids_luma	u(3)
for(i = 0; i < ph_num_alf_ids_luma; i++)	
ph_alf_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
ph_alf_chroma_idc	u(2)
if(ph_alf_chroma_idc > 0)	
ph_alf_id_chroma	u(3)
if(sps_ccalf_enabled_flag) {	
ph_cc_alf_cb_enabled_flag	u(1)
if(ph_cc_alf_cb_enabled_flag)	
ph_cc_alf_cb_aps_id	u(3)
ph_cc_alf_cr_enabled_flag	u(1)
if(ph_cc_alf_cr_enabled_flag)	
ph_cc_alf_cr_aps_id	u(3)
}	
}	

10

20

}	
if(sps_lmcs_enabled_flag) {	
ph_lmcs_enabled_flag	u(1)
if(ph_lmcs_enabled_flag) {	
ph_lmcs_aps_id	u(2)
if(ChromaArrayType != 0)	
ph_chroma_residual_scale_flag	u(1)
}	
}	
if(sps_explicit_scaling_list_enabled_flag) {	
ph_explicit_scaling_list_enabled_flag	u(1)
if(ph_explicit_scaling_list_enabled_flag)	
ph_scaling_list_aps_id	u(3)
}	
if(sps_virtual_boundaries_enabled_flag && !sps_virtual_boundaries_present_flag) {	
ph_virtual_boundaries_present_flag	u(1)
if(ph_virtual_boundaries_present_flag) {	
ph_num_ver_virtual_boundaries	u(2)
for(i = 0; i < ph_num_ver_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_x[i]	u(13)
ph_num_hor_virtual_boundaries	u(2)
for(i = 0; i < ph_num_hor_virtual_boundaries; i++)	
ph_virtual_boundaries_pos_y[i]	u(13)
}	
}	
if(output_flag_present_flag)	
pic_output_flag	u(1)
if(rpl_info_in_ph_flag)	
ref_pic_lists()	
if(partition_constraints_override_enabled_flag)	
partition_constraints_override_flag	u(1)
if(ph_intra_slice_allowed_flag) {	
if(partition_constraints_override_flag) {	
ph_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
ph_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(ph_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
ph_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
ph_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
if(qbt_dual_tree_intra_flag) {	
ph_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
ph_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)

30

40

50

if(ph_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
ph_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
ph_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
if(cu_qp_delta_enabled_flag)	
ph_cu_qp_delta_subdiv_intra_slice	ue(v)
if(pps_cu_chroma_qp_offset_list_enabled_flag)	
ph_cu_chroma_qp_offset_subdiv_intra_slice	ue(v)
}	
if(ph_inter_slice_allowed_flag) {	
if(partition_constraints_override_flag) {	
ph_log2_diff_min_qt_min_cb_inter_slice	ue(v)
ph_max_mtt_hierarchy_depth_inter_slice	ue(v)
if(ph_max_mtt_hierarchy_depth_inter_slice != 0) {	
ph_log2_diff_max_bt_min_qt_inter_slice	ue(v)
ph_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
}	
if(cu_qp_delta_enabled_flag)	
ph_cu_qp_delta_subdiv_inter_slice	ue(v)
if(pps_cu_chroma_qp_offset_list_enabled_flag)	
ph_cu_chroma_qp_offset_subdiv_inter_slice	ue(v)
if(sps_temporal_mvp_enabled_flag) {	
ph_temporal_mvp_enabled_flag	u(1)
if(ph_temporal_mvp_enabled_flag && rpl_info_in_ph_flag) {	
ph_collocated_from_l0_flag	u(1)
if((ph_collocated_from_l0_flag && num_ref_entries[0][RplIdx[0]] > 1) (!ph_collocated_from_l0_flag && num_ref_entries[1][RplIdx[1]] > 1))	
ph_collocated_ref_idx	ue(v)
}	
}	
mvd_l1_zero_flag	u(1)
if(sps_fpel_mmvd_enabled_flag)	
ph_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_pic_present_flag)	
ph_disable_bdof_flag	u(1)
if(sps_dmvr_pic_present_flag)	
ph_disable_dmvr_flag	u(1)

10

20

if(sps_prof_pic_present_flag)	
ph_disable_prof_flag	u(1)
if((pps_weighted_pred_flag pps_weighted_bipred_flag) && wp_info_in_ph_flag)	
pred_weight_table()	
}	
if(qp_delta_info_in_ph_flag)	
ph_qp_delta	se(v)
if(sps_joint_cbr_enabled_flag)	
ph_joint_cbr_sign_flag	u(1)
if(sps_sao_enabled_flag && sao_info_in_ph_flag) {	
ph_sao_luma_enabled_flag	u(1)
if(ChromaArrayType != 0)	
ph_sao_chroma_enabled_flag	u(1)
}	
if(sps_dep_quant_enabled_flag)	
ph_dep_quant_enabled_flag	u(1)
if(sps_sign_data_hiding_enabled_flag && !ph_dep_quant_enabled_flag)	
pic_sign_data_hiding_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag && dbf_info_in_ph_flag) {	
ph_deblocking_filter_override_flag	u(1)
if(ph_deblocking_filter_override_flag) {	
ph_deblocking_filter_disabled_flag	u(1)
if(!ph_deblocking_filter_disabled_flag) {	
ph_beta_offset_div2	se(v)
ph_tc_offset_div2	se(v)
ph_cb_beta_offset_div2	se(v)
ph_cb_tc_offset_div2	se(v)
ph_cr_beta_offset_div2	se(v)
ph_cr_tc_offset_div2	se(v)
}	
}	
}	
if(picture_header_extension_present_flag) {	
ph_extension_length	ue(v)
for(i = 0; i < ph_extension_length; i++)	
ph_extension_data_byte[i]	u(8)
}	
}	

30

40

50

【 0 1 1 4 】

III. 例示的なプロファイル情報

【 0 1 1 5 】

表5は、VVCのようないくつかの関連する例における例示的なプロファイル情報を示す。プロファイル情報は、表5に示されるように、SPS内のprofile_tier_level()に存在し、一般的制約条件情報general_constraint_info()を含んでいてもよい。

【 0 1 1 6 】

表6は、VVCのようないくつかの関連する例における例示的な一般的制約条件情報を示す。一般的制約条件情報におけるイントラのみ制約条件フラグ（たとえば、表6のintra_only_constraint_flag）のような第1のフラグは、プロファイルに適合する画像スライスのスライスタイプがイントラスライスであるかどうかを示すために使用されることができる。第1のフラグが1に等しいことは、プロファイルに適合する画像スライスのスライスタイプがイントラスライスであることを指定する（slice_type = Iスライス）。第1のフラグが0に等しいことは、そのような制約条件を課さない。

表5 SPSにおけるプロファイル情報

【表 5】

profile_tier_level(profileTierPresentFlag, maxNumSubLayersMinus1) {	記述子
if(profileTierPresentFlag) {	
general_profile_idc	u(7)
general_tier_flag	u(1)
general_constraint_info()	
}	
general_level_idc	u(8)
...	
}	

表6 プロファイル情報における一般的制約条件情報

【表 6】

general_constraint_info() {	記述子
general_progressive_source_flag	u(1)
general_interlaced_source_flag	u(1)
general_non_packed_constraint_flag	u(1)
general_frame_only_constraint_flag	u(1)
general_non_projected_constraint_flag	u(1)
intra_only_constraint_flag	u(1)
...	
}	

【 0 1 1 7 】

Main 10プロファイルに適合するビットストリームは、以下の制約条件に従うことがある：（i）参照されるSPSは、chroma_format_idcが0または1に等しい；（ii）参照されるSPSは、0から2の範囲（両端含む）のbit_depth_minus8をもつ；（iii）参照されるSPSは、sps_palette_enabled_flagが0に等しい；（iv）ビューパラメータセット（VPS）（利用可能な場合）および参照されるSPSにおけるiのすべての値について、general_level_idcおよびsublayer_level_idc[i]が255（これはレベル8.5を示す）に等しくない；（v）VVCにおけるMain 10プロファイルについて指定されている階層およびレベル制約条件は、適宜、満たされることができる。

【 0 1 1 8 】

Main 10プロファイルへのビットストリームの適合性は、general_profile_idcが1に等しいことによって示される。特定の階層の特定のレベルでMain 10プロファイルに適合するデコードは、以下の条件のすべてが当てはまるすべてのビットストリームをデコードすることができる：（i）ビットストリームが、Main 10プロファイルに適合することが示される；（ii）ビットストリームが、指定された階層よりも低いか、それに等しい階層に適合することが示される；および（iii）ビットストリームが、レベル8.5ではないが指定されたレベルよりも低いか、それに等しいレベルに適合することが示される。

【 0 1 1 9 】

Main 4:4:4 10プロファイルに適合するビットストリームは、以下の制約条件に従うことがある：(i) 参照されるSPSは、0から3までの範囲（両端含む）のchroma_format_idcをもち、(ii) 参照されるSPSは、0から2までの範囲（両端含む）のbit_depth_minus8をもち、(iii) VPS（利用可能な場合）および参照されるSPSにおけるiのすべての値について、general_level_idcおよびsublayer_level_idc[i]は、255（これはレベル8.5を示す）に等しい；(iv) VVCにおけるMain 4:4:4 10プロファイルについて指定されている階層およびレベル制約条件は、適宜、満たされることができる。

【0120】

Main 4:4:4 10プロファイルへのビットストリームの適合性は、general_profile_idcが2に等しいことによって示される。特定の階層の特定のレベルでMain 4:4:4 10プロファイルに適合するデコーダは、以下の条件のすべてが当てはまるすべてのビットストリームをデコードすることができる：(i) ビットストリームが、Main 4:4:4 10またはMain 10プロファイルに適合することが示される；(ii) ビットストリームが、指定された階層よりも低いか、それに等しい階層に適合することが示される；および(iii) ビットストリームが、レベル8.5ではない、指定されたレベルよりも低いか、それに等しいレベルに適合することが示される。

【0121】

IV. イントラスライスのみを含むビデオ・シーケンスについてのプロファイル情報

【0122】

いくつかの関連した例では、イントラスライスのみが符号化されたビデオ・シーケンスまたはビットストリームに含まれることを示すために、SPSレベルのフラグ（たとえば、sps_inter_allowed_flag）が使用されることができる。このフラグは、冗長性を減らすために、インター符号化に関連した構文要素の信号伝達をスキップするために使用できる。たとえば、sps_inter_allowed_flagが1に等しい場合、インター符号化に関連する構文要素がSPSに存在する可能性がある。sps_inter_allowed_flagが0に等しい場合、SPSにはイントラ符号化に関連した構文要素のみが存在できる。

【0123】

さらに、いくつかの関連した例では、符号化されたビデオ・シーケンスにはイントラスライスのみが含まれることを示すために、PPSレベルのフラグ（たとえば、pps_inter_allowed_flag）が使用されることができる。このフラグは、冗長性を減らすために、インター符号化に関連した構文要素の信号伝達をスキップするために使用できる。たとえば、pps_inter_allowed_flagが1に等しい場合、インター符号化に関連する構文要素がPPSに存在する可能性がある。pps_inter_allowed_flagが0に等しい場合、PPSにはイントラ符号化に関連した構文要素のみが存在できる。

【0124】

本開示は、イントラスライスのみが符号化ビデオ・シーケンスに含まれる場合を示すためにプロファイル情報を使用する方法を含む。

【0125】

本開示の諸側面によれば、符号化されたビデオ・シーケンスにイントラスライスのみが含まれることを示すために、全イントラ・プロファイル（an all intra profile）が使用されることができる。全イントラ・プロファイルでは、このプロファイルに適合するすべてのスライスはイントラ符号化される。全イントラ・プロファイルは、プロファイル識別情報（たとえば、表5のgeneral_profile_idc）のようなプロファイル情報によって示されることができる。

【0126】

符号化ビデオ・シーケンスにイントラスライスのみが含まれることを示すために、静止ピクチャー・プロファイルが使用されることができる。静止ピクチャー・プロファイルでは、このプロファイルに適合するすべてのスライスがイントラ符号化される。静止ピクチャーは、プロファイル識別情報（たとえば、一般プロファイルidc）のようなプロファイル情報によって示されることができる。静止ピクチャー・プロファイルは、カメラによって

10

20

30

40

50

捕捉された静止写真、コンピュータで生成された画像、ビデオ・シーケンスからのスナップショットの抽出などのために使用されることができる。静止ピクチャー・プロファイルは、上述したMain 10プロファイルの機能のサブセットを有することができる。

【0127】

ある実施形態では、静止ピクチャー・プロファイルについて、1ピクチャーのみ制約条件フラグ（たとえば、表7のone_picture_only_constraint_flag）のような第2のフラグがプロファイル情報に含められることができる。1ピクチャーのみ制約条件フラグは、すべてのスライスがイントラ符号化されていて、符号化ビデオ・シーケンス内に1つのピクチャーしかないかどうかを示すことができる。一例では、1ピクチャーのみ制約条件フラグが1に等しいことは、符号化ビデオ・シーケンス内の静止ピクチャー・プロファイルに適合するすべてのスライスがイントラ符号化され（たとえば、slice_type=Iスライス）、符号化ビデオ・シーケンス内に1つのピクチャーのみが存在することを指定する。1ピクチャーのみ制約条件フラグが0に等しいことは、そのような制約条件を課さない。

10

【0128】

ある実施形態では、1ピクチャーのみ制約条件フラグは、プロファイル情報（たとえば、profile_tier_level()）における一般的制約条件情報（たとえば、general_constraint_info()）において信号伝達される。表7は、1ピクチャーのみ制約条件フラグを含む一般的制約条件情報の例を提供する。上述したように、表7の一般的制約条件情報は、表5のprofile_tier_level()のようなプロファイル情報に含められることができる。

表7 1ピクチャーのみ制約条件フラグを含む一般的制約条件情報

20

【表7】

General_constraint_info() {	Descriptor
general_progressive_source_flag	u(1)
general_interlaced_source_flag	u(1)
general_non_packed_constraint_flag	u(1)
general_frame_only_constraint_flag	u(1)
general_non_projected_constraint_flag	u(1)
one_picture_only_constraint_flag	u(1)
intra_only_constraint_flag	u(1)
...	
}	

【0129】

本開示の諸側面によれば、ビットストリーム適合性において、イントラのみ制約条件フラグおよび1ピクチャーのみ制約条件フラグのような第1のフラグと第2のフラグの間で、一つまたは複数の制約条件が適用されることができる。

30

【0130】

ある実施形態では、表7に示されるように、1ピクチャーのみ制約条件フラグは、イントラのみ制約条件フラグの前に信号伝達される。1ピクチャーのみ制約条件フラグが、符号化ビデオ・シーケンス内の静止ピクチャー・プロファイルに適合するすべてのスライスがイントラ符号化され、1つのピクチャーに含まれることを示す場合（たとえば、表7においてone_picture_only_constraint_flag=1）、ビットストリームの適合性に基づいて静止ピクチャー・プロファイルに適合するスライスのスライスタイプがイントラスライスであることを示すために、イントラのみ制約条件フラグを設定することができる（たとえば、表7においてintra_only_constraint_flag=1）。

40

【0131】

ある実施形態では、1ピクチャーのみ制約条件フラグが1に等しいことは、静止ピクチャー・プロファイルに適合するスライスのスライスタイプがイントラスライスであり（たとえば、slice_type=Iスライス）、ビットストリーム内に1つのピクチャーのみが存在することを指定する。1ピクチャーのみ制約条件フラグが0に等しいことは、そのような制約条件を課さない。イントラのみ制約条件フラグが1に等しいことは、スライスのスライスタイプがイントラスライスである（slice_type=Iスライス）ことを指定する。イントラのみ制約条件フラグが0に等しいことは、そのような制約条件を課さない。ビットストリーム適合性に基づき、1ピクチャーのみ制約条件フラグが真である場合、イントラのみ制約

50

条件フラグも真である。

【 0 1 3 2 】

ある実施形態では、1ピクチャーのみの制約条件フラグが1に等しい場合、符号化ビデオ・シーケンスにおける唯一のピクチャーは、瞬時デコード・リフレッシュ (instantaneous decoding refresh、IDR) ピクチャーまたはクリーンランダムアクセス (clean random access、CRA) ピクチャーのような、イントラ・ランダム・アクセス・ピクチャー (intra random access picture、IRAP) であることができる。

【 0 1 3 3 】

ある実施形態では、1ピクチャーのみ制約条件フラグが1に等しい場合、ビデオパラメータセット (VPS) が存在しなくてもよく、符号化ビデオ・シーケンスの層 [レイヤー] の数が1に等しくてもよい。

10

【 0 1 3 4 】

ある実施形態では、1ピクチャーのみ制約条件フラグが1に等しい場合、参照ピクチャーリスト (reference picture list、RPL) およびピクチャー順カウント (picture order count、POC) は、ピクチャーヘッダまたはスライスヘッダに存在しなくてもよい。

【 0 1 3 5 】

ある実施形態では、1ピクチャーのみ制約条件フラグが1に等しい場合、アクセス単位デリミタ (access unit delimiter、AUD) およびストリーム終端 (end of stream、EOS) ネットワーク抽象化層 (network abstraction layer、NAL) 単位は、ビットストリームに存在しなくてもよい。

20

【 0 1 3 6 】

ある実施形態では、全イントラ・プロファイルについて、イントラのみ制約条件フラグは、たとえばイントラのみ制約条件フラグが1に設定されたとき、このプロファイルに適合するすべてのスライスがイントラ符号化されるだけであることを示す。よって、全イントラ・プロファイルにおいては、ビットストリームにはイントラスライスのみが存在することができる。

【 0 1 3 7 】

ある実施形態では、静止ピクチャー・プロファイルについて、1ピクチャーのみ制約条件フラグおよびイントラのみ制約条件フラグの両方が1として設定されることができ、これは、ビットストリームにはイントラスライスのみが存在可能であり、ビットストリームには1つのピクチャーのみが存在可能であることを示す。

30

【 0 1 3 8 】

ある実施形態では、静止ピクチャー・プロファイルについて、1ピクチャーのみ制約条件フラグが1として設定されることができ、これは、ビットストリーム内にはイントラスライスのみが存在可能であり、ビットストリーム内には1つのピクチャーのみが存在可能であることを示す。

【 0 1 3 9 】

本開示の諸側面によれば、イントラのみ制約条件フラグに基づいて、一つまたは複数の非イントラ関連構文要素が除外されてもよい。たとえば、イントラのみ制約条件フラグが存在し、ビットストリーム内のすべてのスライスがイントラ符号化されていることを示す場合 (たとえば、イントラのみ制約条件フラグが1に等しい場合)、非イントラの関連する構文要素は信号伝達されない。

40

【 0 1 4 0 】

ある実施形態では、イントラのみ制約条件フラグが存在し、ビットストリーム内のすべてのスライスがイントラ符号化されていることを示す場合 (たとえば、イントラのみ制約条件フラグが1に等しい場合)、一つまたは複数のフラグが0に設定されることができる。たとえば、前述のsps_inter_allowed_flagとpps_inter_allowed_flagの両方が0に設定されることができる。イントラのみ制約条件フラグが存在しないか、0に等しい場合、そのような制約条件は、sps_inter_allowed_flagやpps_inter_allowed_flagのような前記一つまたは複数のフラグには適用されない。

50

【 0 1 4 1 】

本開示の諸側面によれば、1ピクチャーのみ制約条件フラグに基づいて、一つまたは複数の非イントラ関連構文要素が除外されてもよい。たとえば、1ピクチャーのみ制約条件フラグが存在し、ビットストリーム中のすべてのスライスがイントラ符号化されており、1つのピクチャーのみがビットストリーム中に存在することを示す場合、たとえば、1ピクチャーのみ制約条件フラグが1に等しい場合、非イントラ関連構文要素は信号伝達されない。

【 0 1 4 2 】

ある実施形態では、1ピクチャーのみ制約条件フラグが存在し、ビットストリーム内のすべてのスライスがイントラ符号化され、ビットストリーム内に1つのピクチャーのみが存在することを示す場合、たとえば、1ピクチャーのみ制約条件フラグが1に等しい場合、一つまたは複数のフラグが0に設定されてもよい。たとえば、前述のsps_inter_allowed_flagとpps_inter_allowed_flagの両方が0に設定されることができる。1ピクチャーのみ制約条件フラグが存在しないか、0に等しい場合、そのような制約は、前記一つまたは複数のフラグ、たとえばsps_inter_allowed_flagとpps_inter_allowed_flagには適用されない。

10

【 0 1 4 3 】

本開示の諸側面によれば、符号化ビデオ・シーケンス内のすべてのスライスがイントラ符号化され、1つのピクチャーに含まれることを示すために、第3のフラグが使用されることができる。第3のフラグは、SPSの1ピクチャーのみ存在フラグ (SPS only one picture present flag) であり、前記プロファイル情報とは別個に信号伝達されることができる。たとえば、SPSの1ピクチャーのみ存在フラグは、SPSにおいて信号伝達されることができる。SPSの1ピクチャーのみ存在フラグが1に等しいことは、符号化ビデオ・シーケンスにおけるスライスのスライスタイプがイントラスライスであり (slice_type = Iスライス) であり、シーケンス内に1つのピクチャーしかないことを指定する。SPSの1ピクチャーのみ存在フラグが0に等しいことは、そのような制約条件を課さない。

20

【 0 1 4 4 】

ある実施形態では、1ピクチャーのみ制約条件フラグに基づいて、一つまたは複数の構文要素が除外されてもよい。たとえば、1ピクチャーのみ制約条件フラグが存在し、ビットストリーム中のすべてのスライスがイントラ符号化され、かつ、ビットストリーム中に1つのピクチャーのみが存在することを示す場合、たとえば、1ピクチャーのみ制約条件フラグが1に等しい場合、非イントラ関連の構文要素および/またはPOC値およびRPLに関する構文要素は信号伝達されない。

30

【 0 1 4 5 】

ある実施形態では、1ピクチャーのみ制約条件フラグが存在し、1に等しい場合、SPSの1ピクチャーのみ存在フラグは、ビットストリーム適合性に基づいて、1ピクチャーのみ制約条件フラグと同じ値として設定されることができる。

【 0 1 4 6 】

表8は、1ピクチャー制約条件フラグとイントラのみ制約条件フラグの両方を含む、一般的制約条件情報におけるいくつかの例示的な構文要素を示す。上述のように、表8の一般的制約条件情報は、表5のprofile_tier_level()のようなプロファイル情報に含まれることができる。表8では、1ピクチャー制約条件フラグは、general_one_picture_only_constraint_flagである。general_one_picture_only_constraint_flagが1に等しいことは、ビットストリーム内に1つの符号化されたピクチャーしかないことを指定し、general_one_picture_only_constraint_flagが0に等しいことは、そのような制約条件を課さない。さらに、イントラのみ制約条件フラグはintra_only_constraint_flagである。intra_only_constraint_flagが1に等しいことは、スライスヘッダ内のスライスタイプがイントラスライスである (sh_slice_type = I) ことを指定し、intra_only_constraint_flagが0に等しいことは、そのような制約条件を課さない。general_one_picture_only_constraint_flagが1に等しい場合、intra_only_constraint_flagの値は1として設定される

40

50

ことができる。

表8 一般的制約条件情報構文

【表8】

general_constraint_info() {	記述子
general_non_packed_constraint_flag	u(1)
general_frame_only_constraint_flag	u(1)
general_non_projected_constraint_flag	u(1)
general_one_picture_only_constraint_flag	u(1)
intra_only_constraint_flag	u(1)
max_bitdepth_constraint_idc	u(4)
max_chroma_format_constraint_idc	u(2)
single_layer_constraint_flag	u(1)
all_layers_independent_constraint_flag	u(1)
no_ref_pic_resampling_constraint_flag	u(1)
no_res_change_in_clvs_constraint_flag	u(1)
one_tile_per_pic_constraint_flag	u(1)
pic_header_in_slice_header_constraint_flag	u(1)
one_slice_per_pic_constraint_flag	u(1)
one_subpic_per_pic_constraint_flag	u(1)
no_qtbt_dual_tree_intra_constraint_flag	u(1)
no_partition_constraints_override_constraint_flag	u(1)
no_sao_constraint_flag	u(1)
no_alf_constraint_flag	u(1)
no_ccalf_constraint_flag	u(1)
no_joint_cbr_constraint_flag	u(1)
no_mrl_constraint_flag	u(1)
no_isp_constraint_flag	u(1)
no_mip_constraint_flag	u(1)
no_ref_wraparound_constraint_flag	u(1)
no_temporal_mvp_constraint_flag	u(1)
no_sbtmvp_constraint_flag	u(1)
no_amvr_constraint_flag	u(1)
no_bdof_constraint_flag	u(1)
no_dmvr_constraint_flag	u(1)
no_cclm_constraint_flag	u(1)
no_mts_constraint_flag	u(1)
no_sbt_constraint_flag	u(1)
no_lfst_constraint_flag	u(1)
no_affine_motion_constraint_flag	u(1)
no_mmvd_constraint_flag	u(1)
no_smvd_constraint_flag	u(1)
no_prof_constraint_flag	u(1)
no_bcw_constraint_flag	u(1)

10

20

no_ibc_constraint_flag	u(1)
no_ciip_constraint_flag	u(1)
no_gpm_constraint_flag	u(1)
no_ladf_constraint_flag	u(1)
no_transform_skip_constraint_flag	u(1)
no_bdpcm_constraint_flag	u(1)
no_palette_constraint_flag	u(1)
no_act_constraint_flag	u(1)
no_lmcs_constraint_flag	u(1)
no_cu_qp_delta_constraint_flag	u(1)
no_chroma_qp_offset_constraint_flag	u(1)
no_dep_quant_constraint_flag	u(1)
no_sign_data_hiding_constraint_flag	u(1)
no_tsrc_constraint_flag	u(1)
no_mixed_nalu_types_in_pic_constraint_flag	u(1)
no_trail_constraint_flag	u(1)
no_stsa_constraint_flag	u(1)
no_rasl_constraint_flag	u(1)
no_radl_constraint_flag	u(1)
no_idr_constraint_flag	u(1)
no_cra_constraint_flag	u(1)
no_gdr_constraint_flag	u(1)
no_aps_constraint_flag	u(1)
while(!byte_aligned())	
gci_alignment_zero_bit	f(1)
gci_num_reserved_bytes	u(8)
for(i = 0; i < gci_num_reserved_bytes; i++)	
gci_reserved_byte[i]	u(8)
}	

30

40

【0147】

本開示の諸側面によれば、Main 10またはMain 10静止ピクチャー・プロファイルに適合するビットストリームは、以下の制約条件に従うことができる：(i) Main 10静止ピクチャー・プロファイルに適合するビットストリームでは、ビットストリームは1つのピクチャーのみを含む；(ii) 参照されるSPSは、0または1に等しいsps_chroma_forma

50

t_idcを有する；(iii) 参照されるSPSは、0から2の範囲（両端含む）のsps_bit_depth_minus8を有する；(iv) Main 10静止ピクチャー・プロファイルに適合するビットストリームでは、参照されるSPSは、max_dec_pic_buffering_minus1[sps_max_sublayers]が0に等しい；(v) 参照されるSPSは、sps_palette_enabled_flagが0に等しい；(vi) Main 10静止画プロファイルに適合しないMain 10プロファイルに適合するビットストリームでは、general_level_idcおよびsublayer_level_idc[i]は、参照されるVPS（利用可能な場合）および参照されるSPSにおけるiのすべての値について、255（これはレベル15.5を示す）に等しくない；(vii) VVCにおけるMain 10またはMain 10静止ピクチャー・プロファイルについて指定された階層およびレベルの制約条件が、適宜、満たされることができる。

10

【0148】

Main 10プロファイルへのビットストリームの適合性は、プロファイル識別情報（たとえば、表5のgeneral_profile_idc=1）によって示されることができる。

【0149】

Main 10静止ピクチャー・プロファイルへのビットストリームの適合性は、プロファイル識別情報（たとえば、表5のgeneral_profile_idc=1）とともに、1ピクチャー制約条件フラグ（たとえば、表8のgeneral_one_picture_only_constraint_flag=1）によって示されることができる。

【0150】

なお、Main 10静止ピクチャー・プロファイルへのビットストリームの適合性が上記のように示され、かつ、示されたレベルがレベル15.5でない場合には、Main 10プロファイルへのビットストリームの適合性の指示のための条件も満たされる。

20

【0151】

特定の階層の特定のレベルでMain 10プロファイルに適合するデコーダは、以下の条件のすべてが当てはまるすべてのビットストリームをデコードすることができる：(i) ビットストリームがMain 10またはMain 10静止ピクチャー・プロファイルに適合することが示される；(ii) ビットストリームが指定された階層よりも低いか、それに等しい階層に適合することが示される；および(iii) ビットストリームがレベル15.5ではなく、前記指定されたレベルよりも低いか、それに等しいレベルに適合することが示される。

【0152】

30

特定の階層の特定のレベルでMain 10静止ピクチャー・プロファイルに適合するデコーダは、以下の条件のすべてが当てはまるすべてのビットストリームをデコードすることができる：(i) ビットストリームがMain 10静止ピクチャー・プロファイルに適合することが示される；(ii) ビットストリームが指定された階層よりも低いか、それに等しい階層に適合することが示される；および(iii) ビットストリームがレベル15.5ではなく、前記指定されたレベルよりも低いか、それに等しいレベルに適合することが示される。

【0153】

本開示の諸側面によれば、Main 4:4:4 10またはMain 4:4:4 10静止ピクチャー・プロファイルに適合するビットストリームは、以下の制約条件に従うことができる：(i) Main 4:4:4 10静止ピクチャー・プロファイルに適合するビットストリームでは、ビットストリームは1つのピクチャーのみを含む；(ii) 参照されるSPSは、0から3までの範囲（両端含む）のsps_chroma_format_idcを有する；(iii) 参照されるSPSは、0から2までの範囲（両端含む）のsps_bit_depth_minus8を有する；(iv) Main 4:4:4 10静止ピクチャー・プロファイルに適合するビットストリームでは、参照されるSPSは、max_dec_pic_buffering_minus1[sps_max_sublayers_minus1]が0に等しい；(v) Main 4:4:4 10静止ピクチャー・プロファイルに適合しないMain 4:4:4 10プロファイルに適合するビットストリームでは、general_level_idcおよびsublayer_level_idc[i]は、参照されるVPS（利用可能な場合）および参照されるSPSにおけるiのすべての値について、255（これはレベル15.5を示す）に等しくなく、かつ(vi) VVCにおいてMain 4:4:4 10またはMain 4:4:4 10静止ピクチャー・プロファイルのために指定されている階層お

40

50

よびレベル制約条件が、適宜、満たされることができる。

【0154】

Main 4:4:4 10プロファイルへのビットストリームの適合性は、プロファイル識別情報（たとえば、表5のgeneral_profile_idc=2）によって示される。

【0155】

Main 4:4:4 10静止ピクチャー・プロファイルへのビットストリームの適合性は、プロファイル識別情報（たとえば、表5のgeneral_profile_idc=2）とともに、1ピクチャー制約条件フラグ（たとえば、表8のgeneral_one_picture_only_constraint_flag=1）によって示される。

【0156】

なお、Main 10 4:4:4静止ピクチャー・プロファイルへのビットストリームの適合性が上記のように示され、示されるレベルがレベル15.5でない場合には、Main 10 4:4:4プロファイルへのビットストリームの適合性の指示のための条件も満たされる。

【0157】

特定の階層の特定のレベルでMain 4:4:4 10プロファイルに適合するデコーダは、以下の条件のすべてが当てはまるすべてのビットストリームをデコードすることができる：（i）ビットストリームがMain 4:4:4 10、Main 10、Main 4:4:4 10静止ピクチャーまたはMain 10静止ピクチャー・プロファイルに適合することが示される；（ii）ビットストリームが指定された階層よりも低いか、それに等しい階層に適合することが示される；および（iii）ビットストリームがレベル15.5ではなく、前記指定されたレベルよりも低いか、それに等しいレベルに適合することが示される。

【0158】

特定の階層の特定のレベルでMain 4:4:4 10静止ピクチャー・プロファイルに適合するデコーダは、以下の条件のすべてが当てはまるすべてのビットストリームをデコードすることができる：（i）ビットストリームがMain 4:4:4 10静止ピクチャーまたはMain 10静止ピクチャー・プロファイルに適合することが示される；（ii）ビットストリームが指定された階層よりも低いか、それに等しい階層に適合することが示される；および（iii）ビットストリームがレベル15.5ではなく、前記指定されたレベルよりも低いか、それに等しいレベルに適合することが示される。

【0159】

V. 汎用制約条件フラグのグループ

【0160】

上述の一般的な制約条件情報は、複数の構文要素（たとえば、表8の構文要素）を含むことができる。しかしながら、デコーダは、プロファイルに適合するために、前記複数の構文要素のサブセットをデコードするだけでよい。本開示は、一般的制約条件情報における前記複数の構文要素をグループ化するための方法を含む。前記複数の構文要素のグループ化は、デコーダが一般的制約条件情報のパースを早期に終了することを許容でき、それは、デコード・プロセスを高速化することができる。

【0161】

本開示の諸側面によれば、一般的制約条件情報における前記複数の構文要素は、プロファイルのような使用シナリオに基づいてグループ化できる。グループ内のサブグループも存在する可能性がある。構文要素の各グループまたはサブグループは、ビットストリームにおいて連続して存在する構文要素のトランク（trunk）を含む。よって、デコーダは、グループの知識を用いて、一般的制約条件情報のパースを早期に終了することができる。

【0162】

いくつかの実施形態では、デコーダは、一般的制約条件情報に含まれる構文要素のグループの総数および順序の知識を有することができる。

【0163】

ある実施形態では、表9に示されるように、一般的制約条件情報は、構文要素の3つのグループ、すなわち、非イントラ非インター・グループ（グループI）、イントラ・グループ

10

20

30

40

50

(グループII)、およびインター・グループ(グループIII)を含む。たとえば、イントラ・グループ(グループII)はイントラ符号化ツールに関連する構文要素のトランクを含み、インター・グループ(グループIII)はインター符号化ツールに関連する構文要素のトランクを含み、非イントラ非インター・グループ(グループI)はイントラ符号化ツールにもインター符号化ツールも関連しない構文要素のトランクを含む。一例では、非イントラ非インター・グループ(グループI)が最初にビットストリームに存在し、次いでイントラ・グループ(グループII)が続き、インター・グループ(グループIII)が最後に現れる。

【 0 1 6 4 】

上述のように、表9における一般的制約条件情報は、表5のprofile_tier_level()のようなプロファイル情報に含まれることができる。

表9 構文要素の3つのグループを含む一般的制約条件情報

【表 9】

general_constraint_info() {	記述子
//グループI: 非イントラ非インター・グループ	
general_non_packed_constraint_flag	u(1)
general_frame_only_constraint_flag	u(1)
general_non_projected_constraint_flag	u(1)
general_one_picture_only_constraint_flag	u(1)
intra_only_constraint_flag	u(1)
max_bitdepth_constraint_idc	u(4)
max_chroma_format_constraint_idc	u(2)
single_layer_constraint_flag	u(1)
all_layers_independent_constraint_flag	u(1)
no_ref_pic_resampling_constraint_flag	u(1)
no_res_change_in_clvs_constraint_flag	u(1)
one_tile_per_pic_constraint_flag	u(1)
pic_header_in_slice_header_constraint_flag	u(1)
one_slice_per_pic_constraint_flag	u(1)
one_subpic_per_pic_constraint_flag	u(1)
no_partition_constraints_override_constraint_flag	u(1)
no_sao_constraint_flag	u(1)
no_alf_constraint_flag	u(1)
no_ccalf_constraint_flag	u(1)
no_joint_cbr_constraint_flag	u(1)
no_mts_constraint_flag	u(1)
no_ladf_constraint_flag	u(1)
no_transform_skip_constraint_flag	u(1)
no_act_constraint_flag	u(1)
no_lmcs_constraint_flag	u(1)
no_cu_qp_delta_constraint_flag	u(1)
no_chroma_qp_offset_constraint_flag	u(1)
no_dep_quant_constraint_flag	u(1)
no_sign_data_hiding_constraint_flag	u(1)
no_tsrc_constraint_flag	u(1)
no_mixed_nalu_types_in_pic_constraint_flag	u(1)
no_trail_constraint_flag	u(1)
no_stsa_constraint_flag	u(1)
no_rasl_constraint_flag	u(1)
no_radl_constraint_flag	u(1)
no_idr_constraint_flag	u(1)
no_cra_constraint_flag	u(1)

10

20

30

40

50

no_gdr_constraint_flag	u(1)
no_aps_constraint_flag	u(1)
//グループI: 非イントラ非インター・グループの終わり	
//グループII: イントラ・グループ	
no_qtbt_dual_tree_intra_constraint_flag	u(1)
no_mrl_constraint_flag	u(1)
no_isp_constraint_flag	u(1)
no_mip_constraint_flag	u(1)
no_cclm_constraint_flag	u(1)
no_lfst_constraint_flag	u(1)
no_ibc_constraint_flag	u(1)
no_bdpcm_constraint_flag	u(1)
no_palette_constraint_flag	u(1)
//グループII: イントラ・グループの終わり	
//グループIII: インター・グループ	
no_ref_wraparound_constraint_flag	u(1)
no_temporal_mvp_constraint_flag	u(1)
no_sbtmvp_constraint_flag	u(1)
no_amvr_constraint_flag	u(1)
no_bdof_constraint_flag	u(1)
no_dmvr_constraint_flag	u(1)
no_sbt_constraint_flag	u(1)
no_affine_motion_constraint_flag	u(1)
no_mmvd_constraint_flag	u(1)
no_smvd_constraint_flag	u(1)
no_prof_constraint_flag	u(1)
no_bcw_constraint_flag	u(1)
no_ciip_constraint_flag	u(1)
no_gpm_constraint_flag	u(1)
//グループIII: インター・グループの終わり	
while(!byte_aligned())	
gci_alignment_zero_bit	f(1)
gci_num_reserved_bytes	u(8)
for(i = 0; i < gci_num_reserved_bytes; i++)	
gci_reserved_byte[i]	u(8)
}	

10

20

【 0 1 6 5 】

Main 10 静止ピクチャーまたはMain 10 4:4:4 静止ピクチャー・プロファイルでは、ビットストリームの適合性は、プロファイル識別情報（たとえば、表5のgeneral_profile_idc = 1）とともに、1ピクチャーのみ制約条件フラグ（たとえば、表9のgeneral_one_picture_only_constraint_flag = 1）によって示されることができる。1ピクチャーのみ制約条件フラグが1に等しい場合、イントラのみ制約条件フラグの値は1に等しい。他のプロファイル、たとえばイントラ・ピクチャーのみを含むプロファイルでは、イントラのみ制約条件フラグも1に等しくてもよい。

30

【 0 1 6 6 】

イントラのみ制約条件フラグが1に等しい場合、表9のno_ref_wraparound_constraint_flag、no_temporal_mvp_constraint_flagなどのインター・グループ（グループIII）における構文要素の値はすべて1に等しい。

【 0 1 6 7 】

よって、構文要素のグループの知識により、デコーダは、一般的制約条件情報のパースを早期に終了することができ、これは、デコード速度および手順に有益でありうる。

【 0 1 6 8 】

ある実施形態では、表10に示されるように、一般的制約条件情報は、構文要素の2つのグループ、すなわち、非インター・グループ（グループI）およびインター・グループ（グループII）を含む。インター・グループ（グループII）は、インター符号化ツールに関連する構文要素のトランクを含むことができる。非インター・グループ（グループI）は、インター符号化ツールに関連しない構文要素のトランクを含むことができる。一例では、非インター・グループ（グループI）が最初にビットストリームに現れ、インター・グループ（グループII）がそれに続く。

40

【 0 1 6 9 】

上述のように、表10の一般的制約条件情報は、表5のprofile_tier_level()のようなプロファイル情報に含まれることができる。

表10 構文要素の2つのグループを含む一般的制約条件情報

50

【表 10】

general_constraint_info() {	記述子
//グループI: 非インター・グループ	
general_non_packed_constraint_flag	u(1)
general_frame_only_constraint_flag	u(1)
general_non_projected_constraint_flag	u(1)
general_one_picture_only_constraint_flag	u(1)
intra_only_constraint_flag	u(1)
max_bitdepth_constraint_idc	u(4)
max_chroma_format_constraint_idc	u(2)
single_layer_constraint_flag	u(1)
all_layers_independent_constraint_flag	u(1)
no_ref_pic_resampling_constraint_flag	u(1)
no_res_change_in_clvs_constraint_flag	u(1)
one_tile_per_pic_constraint_flag	u(1)
pic_header_in_slice_header_constraint_flag	u(1)
one_slice_per_pic_constraint_flag	u(1)
one_subpic_per_pic_constraint_flag	u(1)
no_partition_constraints_override_constraint_flag	u(1)
no_sao_constraint_flag	u(1)
no_alf_constraint_flag	u(1)
no_ccalf_constraint_flag	u(1)
no_joint_cbr_constraint_flag	u(1)
no_mts_constraint_flag	u(1)
no_ladf_constraint_flag	u(1)
no_transform_skip_constraint_flag	u(1)
no_act_constraint_flag	u(1)
no_lmcs_constraint_flag	u(1)
no_cu_qp_delta_constraint_flag	u(1)
no_chroma_qp_offset_constraint_flag	u(1)
no_dep_quant_constraint_flag	u(1)
no_sign_data_hiding_constraint_flag	u(1)
no_tsrc_constraint_flag	u(1)
no_mixed_nalu_types_in_pic_constraint_flag	u(1)
no_trail_constraint_flag	u(1)
no_stsa_constraint_flag	u(1)
no_rasl_constraint_flag	u(1)
no_radl_constraint_flag	u(1)
no_idr_constraint_flag	u(1)
no_cra_constraint_flag	u(1)

10

20

no_gdr_constraint_flag	u(1)
no_aps_constraint_flag	u(1)
no_qtbt_dual_tree_intra_constraint_flag	u(1)
no_mrl_constraint_flag	u(1)
no_isp_constraint_flag	u(1)
no_mip_constraint_flag	u(1)
no_cclm_constraint_flag	u(1)
no_lfst_constraint_flag	u(1)
no_ibc_constraint_flag	u(1)
no_bdpcm_constraint_flag	u(1)
no_palette_constraint_flag	u(1)
//グループI: 非インター・グループの終わり	
//グループII: インター・グループ	
no_ref_wraparound_constraint_flag	u(1)
no_temporal_mvp_constraint_flag	u(1)
no_sbtmvp_constraint_flag	u(1)
no_amvr_constraint_flag	u(1)
no_bdof_constraint_flag	u(1)
no_dmvr_constraint_flag	u(1)
no_sbt_constraint_flag	u(1)
no_affine_motion_constraint_flag	u(1)
no_mmvd_constraint_flag	u(1)
no_smvd_constraint_flag	u(1)
no_prof_constraint_flag	u(1)
no_bcw_constraint_flag	u(1)
no_ciip_constraint_flag	u(1)
no_gpm_constraint_flag	u(1)
//グループII: インター・グループの終わり	
while(!byte_aligned())	
gci_alignment_zero_bit	f(1)
gci_num_reserved_bytes	u(8)
for(i = 0; i < gci_num_reserved_bytes; i++)	
gci_reserved_byte[i]	u(8)
}	

30

40

【0170】

Main 10 静止ピクチャーまたはMain 10 4:4:4 静止ピクチャー・プロファイルでは、ビットストリームの適合性は、プロファイル識別情報（表5のgeneral_profile_idc = 1）とともに、1ピクチャーのみ制約条件フラグ（たとえば、表10のgeneral_one_picture

50

_only_constraint_flag = 1) によって示されることができる。1ピクチャーのみの制約条件フラグが1に等しい場合、イントラのみ制約条件フラグの値は1に等しい。他のプロファイル、たとえばイントラ・ピクチャーのみを含むプロファイルでは、イントラのみ制約条件フラグも1に等しくてもよい。

【0171】

イントラのみ制約条件フラグが1に等しい場合、表10のno_ref_wraparound_constraint_flag、no_temporal_mvp_constraint_flagなどのインター・グループ（グループII）の構文要素の値はすべて1に等しい。

【0172】

よって、構文要素のグループの知識により、デコーダは、一般的制約条件情報のパースを早期に終了することができ、これはデコード速度および手順に有益である。

10

【0173】

いくつかの実施形態によれば、新しい構文要素が一般的制約条件情報（たとえば、表9または表10のgeneral_constraint_info()）に導入されるときはいつでも、一般的制約条件情報が既存のグループに対して新しいグループを形成する必要があるように、新しい構文要素が既存のグループまたはサブグループに含まれることができる。

【0174】

ある実施形態では、非イントラ非インター・グループ（グループI）、イントラ・グループ（グループII）、およびインター・グループ（グループIII）を含む一般的制約条件情報について、イントラ符号化ツールに関連する新しい構文要素が一般的制約条件情報に含められる場合、新しい構文要素は、イントラ・グループ（グループII）に含められることができる。インター符号化ツールに関連する新しい構文要素が一般的制約条件情報に含められる場合、新しい構文要素は、インター・グループ（グループIII）に含められることができる。新しい構文要素がイントラ符号化にもインター符号化ツールにも関連しない場合、新しい構文要素は、非イントラ非インター・グループ（グループI）に含められることができる。

20

【0175】

表11に示される例では、重み付け予測がPスライスに適用できるかどうかを指定する新しい構文要素、重み付け予測なし制約条件フラグ（たとえば、no_weighted_pred_constraint_flag）が、構文要素の3つのグループを含む一般的制約条件情報に導入される。新しい構文要素は、インター符号化ツールに関連する構文要素であるので、インター・グループ（グループIII）に含められる。

30

【0176】

上述したように、表11の一般的制約条件情報は、表5のprofile_tier_level()のようなプロファイル情報に含められることができる。

表11 構文要素の3つのグループを含む一般的制約条件情報に導入する新しい構文要素

40

【表 1 1】

general_constraint_info() {	記述子
//グループI:非イントラ非インター・グループ	
...	
//グループI:非イントラ非インター・グループの終わり	
//グループII:イントラ・グループ	
...	
//グループII:イントラ・グループの終わり	
//グループIII:インター・グループ	
no_ref_wraparound_constraint_flag	u(1)
no_temporal_mvp_constraint_flag	u(1)
no_sbtmvp_constraint_flag	u(1)
no_amvr_constraint_flag	u(1)
no_bdof_constraint_flag	u(1)
no_dmvr_constraint_flag	u(1)
no_sbt_constraint_flag	u(1)
no_affine_motion_constraint_flag	u(1)
no_mmvd_constraint_flag	u(1)
no_smvd_constraint_flag	u(1)
no_prof_constraint_flag	u(1)
no_bcw_constraint_flag	u(1)
no_cip_constraint_flag	u(1)
no_gpm_constraint_flag	u(1)
no_weighted_pred_constraint_flag	u(1)
//end of Group III inter group	
while(!byte_aligned())	
gci_alignment_zero_bit	f(1)
gci_num_reserved_bytes	u(8)
for(i = 0; i < gci_num_reserved_bytes; i++)	
gci_reserved_byte[i]	u(8)
}	

10

【 0 1 7 7 】

20

ある実施形態では、非インター・グループ（グループI）およびインター・グループ（グループII）を含む一般的制約条件情報について、インター符号化ツールに関連する新しい構文要素が一般的制約条件情報に含められる場合、新しい構文はインター・グループ（グループII）に含められることができる。インター符号化ツールに関係しない新しい構文要素が一般的制約条件情報に含められる場合、新しい構文要素は、非インターグループ（グループI）に含められることができる。

【 0 1 7 8 】

表12に示される例では、重み付け予測がPスライスに適用できるかどうかを指定する新しい構文要素、重み付け予測なし制約条件フラグ（たとえば、no_weighted_pred_constraint_flag）が、構文要素の2つのグループを含む一般的制約条件情報に導入される。新しい構文要素は、インター符号化ツールに関連する構文要素であるので、インター・グループ（グループII）に含められることができる。

30

【 0 1 7 9 】

上述したように、表12の一般的制約条件情報は、表5のprofile_tier_level()のようなプロファイル情報に含められることができる。

表12 構文要素の2つのグループを含む一般的制約条件情報に導入する新しい構文要素

40

50

【表 1 2】

general_constraint_info() {	記述子
//グループ1: 他のグループ	
...	
//グループ1: 他のグループの終わり	
//グループ2: インターグループ	
no_ref_wraparound_constraint_flag	u(1)
no_temporal_mvp_constraint_flag	u(1)
no_sbtmvp_constraint_flag	u(1)
no_amvr_constraint_flag	u(1)
no_bdof_constraint_flag	u(1)
no_dmvr_constraint_flag	u(1)
no_sbt_constraint_flag	u(1)
no_affine_motion_constraint_flag	u(1)
no_mnvd_constraint_flag	u(1)
no_smvd_constraint_flag	u(1)
no_prof_constraint_flag	u(1)
no_bcw_constraint_flag	u(1)
no_cup_constraint_flag	u(1)
no_gpm_constraint_flag	u(1)
no_weighted_pred_constraint_flag	u(1)
//グループ2: インターグループの終わり	
while('byte_aligned())	
gci_alignment_zero_bit	f(1)
gci_num_reserved_bytes	u(8)
for(i = 0; i < gci_num_reserved_bytes; i++)	
gci_reserved_byte[i]	u(8)
}	

10

【 0 1 8 0 】

本開示の諸側面によれば、構文解析および早期終了を容易にするために、各グループまたはサブグループの後にバイト整列がチェックされることができる。

20

【 0 1 8 1 】

表13は、構文要素の3つのグループを含む一般的制約条件情報における各グループのための例示的なバイト整列を示す。バイト整列は、各グループまたはサブグループの末尾でチェックされる。グループまたはサブグループ内で信号伝達される構文要素がバイト整列されていない場合、各グループについて使用される総ビットがバイト整列されることを保証するために、追加的な諸ビットが信号伝達されることができる。

【 0 1 8 2 】

上述したように、表13の一般的制約条件情報は、表5のprofile_tier_level()のようなプロファイル情報に含められることができる。

30

表13 構文要素の3つのグループを含む一般的制約条件情報における各グループについてのバイト整列

40

50

【表 1 3】

general_constraint_info() {	記述子
//グループI: 非イントラ非インター・グループ	
general_non_packed_constraint_flag	u(1)
general_frame_only_constraint_flag	u(1)
general_non_projected_constraint_flag	u(1)
general_one_picture_only_constraint_flag	u(1)
intra_only_constraint_flag	u(1)
max_bitdepth_constraint_idc	u(4)
max_chroma_format_constraint_idc	u(2)
single_layer_constraint_flag	u(1)
all_layers_independent_constraint_flag	u(1)
no_ref_pic_resampling_constraint_flag	u(1)
no_res_change_in_clvs_constraint_flag	u(1)
one_tile_per_pic_constraint_flag	u(1)
pic_header_in_slice_header_constraint_flag	u(1)
one_slice_per_pic_constraint_flag	u(1)
one_subpic_per_pic_constraint_flag	u(1)
no_partition_constraints_override_constraint_flag	u(1)
no_sao_constraint_flag	u(1)
no_alf_constraint_flag	u(1)
no_ccalf_constraint_flag	u(1)
no_joint_cber_constraint_flag	u(1)
no_mts_constraint_flag	u(1)
no_ladf_constraint_flag	u(1)
no_transform_skip_constraint_flag	u(1)
no_act_constraint_flag	u(1)
no_lmcs_constraint_flag	u(1)
no_cu_qp_delta_constraint_flag	u(1)
no_chroma_qp_offset_constraint_flag	u(1)
no_dep_quant_constraint_flag	u(1)
no_sign_data_hiding_constraint_flag	u(1)
no_tsrc_constraint_flag	u(1)
no_mixed_nalu_types_in_pic_constraint_flag	u(1)
no_trail_constraint_flag	u(1)
no_stsa_constraint_flag	u(1)
no_rasl_constraint_flag	u(1)
no_radl_constraint_flag	u(1)
no_idr_constraint_flag	u(1)
no_cra_constraint_flag	u(1)

10

20

no_gdr_constraint_flag	u(1)
no_aps_constraint_flag	u(1)
while(!byte_aligned())	
gci_alignment_zero_bit	fl(1)
//グループI: 非イントラ非インター・グループの終わり	
//グループII: イントラ・グループ	
no_qtbt_dual_tree_intra_constraint_flag	u(1)
no_mrl_constraint_flag	u(1)
no_isp_constraint_flag	u(1)
no_mip_constraint_flag	u(1)
no_cclm_constraint_flag	u(1)
no_lfnst_constraint_flag	u(1)
no_ibc_constraint_flag	u(1)
no_bdpcm_constraint_flag	u(1)
no_palette_constraint_flag	u(1)
while(!byte_aligned())	
gci_alignment_zero_bit	fl(1)
//グループII: イントラ・グループの終わり	
//グループIII: インター・グループ	
no_ref_wraparound_constraint_flag	u(1)
no_temporal_mvp_constraint_flag	u(1)
no_sbtmvp_constraint_flag	u(1)
no_amvr_constraint_flag	u(1)
no_bdof_constraint_flag	u(1)
no_dmvr_constraint_flag	u(1)
no_sbt_constraint_flag	u(1)
no_affine_motion_constraint_flag	u(1)
no_mmvd_constraint_flag	u(1)
no_smvd_constraint_flag	u(1)
no_prof_constraint_flag	u(1)
no_bcw_constraint_flag	u(1)
no_ciip_constraint_flag	u(1)
no_gpm_constraint_flag	u(1)
while(!byte_aligned())	
gci_alignment_zero_bit	fl(1)
//グループIII: インター・グループの終わり	
gci_num_reserved_bytes	u(8)
for(i = 0; i < gci_num_reserved_bytes; i++)	
gci_reserved_byte[i]	u(8)
}	

30

40

50

【 0 1 8 3 】

表14は、構文要素の2つのグループを含む一般的制約条件情報における各グループについての例示的なバイト整列を示す。バイト整列は、各グループまたはサブグループの末尾でチェックされる。グループまたはサブグループ内で信号伝達される構文要素がバイト整列されていない場合、各グループについて使用される総ビットがバイト整列されることを保証するために、追加的な諸ビットが信号伝達される。

【 0 1 8 4 】

上述したように、表14の一般的制約条件情報は、表5のprofile_tier_level()のようなプロファイル情報に含められることができる。

表14 構文要素の2つのグループを含む一般的制約条件情報における各グループについてのバイト整列

【表 1 4】

general_constraint_info() {	記述子
//グループ1:他のグループ	
general_non_packed_constraint_flag	u(1)
general_frame_only_constraint_flag	u(1)
general_non_projected_constraint_flag	u(1)
general_one_picture_only_constraint_flag	u(1)
intra_only_constraint_flag	u(1)
max_bitdepth_constraint_idc	u(4)
max_chroma_format_constraint_idc	u(2)
single_layer_constraint_flag	u(1)
all_layers_independent_constraint_flag	u(1)
no_ref_pic_resampling_constraint_flag	u(1)
no_res_change_in_clvs_constraint_flag	u(1)
one_tile_per_pic_constraint_flag	u(1)
pic_header_in_slice_header_constraint_flag	u(1)
one_slice_per_pic_constraint_flag	u(1)
one_subpic_per_pic_constraint_flag	u(1)
no_partition_constraints_override_constraint_flag	u(1)
no_sao_constraint_flag	u(1)
no_alf_constraint_flag	u(1)
no_ccalf_constraint_flag	u(1)
no_joint_cbr_constraint_flag	u(1)
no_mts_constraint_flag	u(1)
no_ladf_constraint_flag	u(1)
no_transform_skip_constraint_flag	u(1)
no_act_constraint_flag	u(1)
no_lmcs_constraint_flag	u(1)
no_cu_qp_delta_constraint_flag	u(1)
no_chroma_qp_offset_constraint_flag	u(1)
no_dep_quant_constraint_flag	u(1)
no_sign_data_hiding_constraint_flag	u(1)
no_tsrc_constraint_flag	u(1)
no_mixed_nalu_types_in_pic_constraint_flag	u(1)
no_trail_constraint_flag	u(1)
no_stsa_constraint_flag	u(1)
no_rasl_constraint_flag	u(1)
no_radl_constraint_flag	u(1)
no_idr_constraint_flag	u(1)
no_cra_constraint_flag	u(1)

10

20

30

40

50

no_gdr_constraint_flag	u(1)
no_aps_constraint_flag	u(1)
no_qbtt_dual_tree_intra_constraint_flag	u(1)
no_nrl_constraint_flag	u(1)
no_isp_constraint_flag	u(1)
no_mip_constraint_flag	u(1)
no_cclm_constraint_flag	u(1)
no_lfst_constraint_flag	u(1)
no_ibc_constraint_flag	u(1)
no_bdpcm_constraint_flag	u(1)
no_palette_constraint_flag	u(1)
while(!byte_aligned())	
gci_alignment_zero_bit	fi(1)
//グループI:非インター・グループの終わり	
//グループII:インター・グループ	
no_ref_wraparound_constraint_flag	u(1)
no_temporal_mvp_constraint_flag	u(1)
no_sbtmvp_constraint_flag	u(1)
no_amvr_constraint_flag	u(1)
no_bdof_constraint_flag	u(1)
no_dmvr_constraint_flag	u(1)
no_sbt_constraint_flag	u(1)
no_affine_motion_constraint_flag	u(1)
no_mmvd_constraint_flag	u(1)
no_smvd_constraint_flag	u(1)
no_prof_constraint_flag	u(1)
no_bcw_constraint_flag	u(1)
no_ciip_constraint_flag	u(1)
no_gpm_constraint_flag	u(1)
while(!byte_aligned())	
gci_alignment_zero_bit	fi(1)
//グループII:インター・グループの終わり	
gci_num_reserved_bytes	u(8)
for(i=0; i<gci_num_reserved_bytes; i++)	
gci_reserved_byte[i]	u(8)
}	

10

20

【 0 1 8 5 】

VII. フローチャート

【 0 1 8 6 】

図8は、本開示のある実施形態による例示的なプロセス（800）の概略を示すフローチャートを示す。さまざまな実施形態において、プロセス（800）は、端末装置（210）、（220）、（230）および（240）内の処理回路、ビデオエンコーダ（303）の機能を実行する処理回路、ビデオデコーダ（310）の機能を実行する処理回路、ビデオデコーダ（410）の機能を実行する処理回路、イントラ予測モジュール（452）の機能を実行する処理回路、ビデオエンコーダ（503）の機能を実行する処理回路、予測器（535）の機能を実行する処理回路、イントラエンコーダ（622）の機能を実行する処理回路、イントラ・デコーダ（772）の機能を実行する処理回路などの処理回路などによって実行される。いくつかの実施形態では、プロセス（800）は、ソフトウェア命令で実装され、よって、処理回路がソフトウェア命令を実行すると、処理回路は、プロセス（800）を実行する。

30

【 0 1 8 7 】

プロセス（800）は、一般に、ステップ（S810）で開始され、プロセス（800）は、符号化ビデオ・ビットストリームについての予測情報における複数の画像スライスについてのプロファイル情報をデコードする。プロファイル情報は、符号化ビデオ・ビットストリーム内の画像スライスのそれぞれがイントラ符号化されるプロファイルのプロファイル識別情報を含む。次いで、プロセス（800）は、ステップ（S820）に進む。

40

【 0 1 8 8 】

ステップ（S820）において、プロセス（800）は、符号化ビデオ・ビットストリーム内の各画像スライスに対してイントラ予測を実行する。次いで、プロセス（800）は、ステップ（S830）に進む。

【 0 1 8 9 】

ステップ（S830）において、プロセス（800）は、イントラ予測に基づいて少なくとも1つの画像ピクチャーを再構成する。次いで、プロセス（800）は終了する。

【 0 1 9 0 】

50

ある実施形態では、プロファイル情報は、符号化ビデオ・ビットストリーム内の各画像スライスがイントラ符号化されているかどうかを示す第1のフラグと、符号化ビデオ・ビットストリーム内の各画像スライスが1つのピクチャーに含まれているかどうかを示す第2のフラグを含む。

【0191】

ある実施形態では、第1のフラグは、第2のフラグの後にデコードされ、第2のフラグが符号化ビデオ・ビットストリーム内の各画像スライスが1つのピクチャーに含まれることを示すことに基づいて、符号化ビデオ・ビットストリーム内の各画像スライスがイントラ符号化されていることを示す。

【0192】

ある実施形態では、第1のフラグは、符号化ビデオ・ビットストリーム内の各画像スライスがイントラ符号化されるプロファイルのプロファイル識別情報に基づいて、符号化ビデオ・ビットストリーム内の各画像スライスがイントラ符号化されていることを示す。

【0193】

ある実施形態では、第2のフラグは、前記プロファイルが符号化ビデオ・ビットストリーム内に1つのピクチャーのみが含まれる静止ピクチャー・プロファイルであることに基づいて、符号化ビデオ・ビットストリーム内の各画像スライスが1つのピクチャーに含まれることを示す。

【0194】

ある実施形態では、非イントラ関連構文要素は、(i) 第1のフラグが、符号化ビデオ・ビットストリーム内の各画像スライスがイントラ符号化されることを示すこと、(ii) 第2のフラグが、符号化ビデオ・ビットストリーム内の各画像スライスが1つのピクチャーに含まれることを示すことのうちの1つに基づいて、前記予測情報に含まれない。

【0195】

ある実施形態では、前記予測情報は、符号化ビデオ・ビットストリーム内の各画像スライスがイントラ符号化され、1つのピクチャーに含まれるかどうかを示す第3のフラグを含む。第3のフラグは、前記プロファイル情報には含まれない。

【0196】

ある実施形態では、第3のフラグは、第2のフラグが、符号化ビデオ・ビットストリーム内の各画像スライスが1つのピクチャーに含まれることを示すことに基づいて、符号化ビデオ・ビットストリーム内の各画像スライスがイントラ符号化され、1つのピクチャーに含まれることを示す。

【0197】

図9は、本開示のある実施形態による例示的なプロセス(900)の概略を示す別のフローチャートを示す。さまざまな実施形態において、プロセス(900)は、端末装置(210)、(220)、(230)および(240)内の処理回路、ビデオエンコーダ(303)の機能を実行する処理回路、ビデオデコーダ(310)の機能を実行する処理回路、ビデオデコーダ(410)の機能を実行する処理回路、イントラ予測モジュール(452)の機能を実行する処理回路、ビデオエンコーダ(503)の機能を実行する処理回路、予測器(535)の機能を実行する処理回路、イントラエンコーダ(622)の機能を実行する処理回路、イントラ・デコーダ(772)の機能を実行する処理回路などの処理回路などによって実行される。いくつかの実施形態では、プロセス(900)は、ソフトウェア命令で実装され、よって、処理回路がソフトウェア命令を実行すると、処理回路は、プロセス(900)を実行する。

【0198】

プロセス(900)は、一般に、ステップ(S910)で開始されてもよく、ここで、プロセス(900)は、符号化ビデオ・ビットストリームの予測情報におけるプロファイル情報をデコードする。プロファイル情報は、構文要素の複数のグループを含み、符号化ビデオ・ビットストリームのプロファイルを示す。次いで、プロセス(900)は、ステップ(S920)に進む。

10

20

30

40

50

【0199】

ステップ(S920)において、プロセス(900)は、プロファイル情報において示されたプロファイルに基づいて、構文要素の複数のグループのうちの少なくとも1つを決定する。次いで、プロセス(900)は、ステップ(S930)に進む。

【0200】

ステップ(S930)において、プロセス(900)は、構文要素の前記複数のグループのうちの前記決定された少なくとも1つに基づいて、前記予測情報に含まれる構文要素をデコードする。次いで、プロセス(900)は、ステップ(S940)に進む。

【0201】

ステップ(S940)において、プロセス(900)は、予測情報に含まれるデコードされた構文要素に基づいて、少なくとも1つのピクチャーを再構成する。

10

【0202】

ある実施形態では、プロファイルについての構文要素の前記複数のグループのうちの前記決定された少なくとも1つのグループの順序は、前記プロファイル情報における構文要素の前記複数のグループのあらかじめ決定された順序に従う。

【0203】

ある実施形態では、バイト整列は、プロファイル情報内の構文要素の前記複数のグループのそれぞれについてチェックされる。

【0204】

VIII. コンピュータ・システム

20

【0205】

上述の技法は、コンピュータ読み取り可能な命令を用いてコンピュータ・ソフトウェアとして実装することができ、一つまたは複数のコンピュータ読み取り可能な媒体に物理的に記憶されることができる。たとえば、図10は、開示された主題のある種の実施形態を実施するのに好適なコンピュータ・システム(1000)を示す。

【0206】

コンピュータ・ソフトウェアは、任意の好適な機械コードまたはコンピュータ言語を用いてコーディングされることができ、アセンブリ、コンパイル、リンク、または同様の機構の対象とされて、一つまたは複数のコンピュータ中央処理ユニット(CPU)、グラフィックス処理ユニット(GPU)などによって、直接的に、またはインタープリット、マイクロコード実行などを通じて実行可能な命令を含むコードを作成することができる。

30

【0207】

命令は、たとえば、パーソナルコンピュータ、タブレット・コンピュータ、サーバー、スマートフォン、ゲーム装置、モノのインターネット装置等を含むさまざまなタイプのコンピュータまたはそのコンポーネント上で実行されることができる。

【0208】

コンピュータ・システム(1000)について図10に示されるコンポーネントは、例としての性質であり、本開示の実施形態を実装するコンピュータ・ソフトウェアの使用または機能の範囲に関する制限を示唆することを意図したものではない。コンポーネントの構成も、コンピュータ・システム(1000)の例示的实施形態において示されているコンポーネントの任意の1つまたは組み合わせに関する何らかの依存性または要件を有するものとして解釈されるべきではない。

40

【0209】

コンピュータ・システム(1000)は、ある種のヒューマン・インターフェース入力装置を含むことができる。そのようなヒューマン・インターフェース入力装置は、たとえば、触覚入力(たとえば、キーストローク、スワイプ、データグローブの動き)、音声入力(たとえば、声、拍手)、視覚入力(たとえば、ジェスチャー)、嗅覚入力(図示せず)を通じた一または複数の人間ユーザーによる入力に応答することができる。また、ヒューマン・インターフェース装置は、音声(たとえば、発話、音楽、周囲の音)、画像(たとえば、スキャンされた画像、スチール画像カメラから得られる写真画像)、ビデオ(たと

50

えば、2次元ビデオ、立体視ビデオを含む3次元ビデオ)のような、人間による意識的入力に必ずしも直接関係しないある種のメディアを捕捉するために使用できる。

【0210】

入力ヒューマン・インターフェース装置は、キーボード(1001)、マウス(1002)、トラックパッド(1003)、タッチスクリーン(1010)、データグローブ(図示せず)、ジョイスティック(1005)、マイクロフォン(1006)、スキャナ(1007)、カメラ(1008)の一つまたは複数(それぞれの一つしか図示していない)を含んでいてもよい。

【0211】

コンピュータ・システム(1000)はまた、ある種のヒューマン・インターフェース出力装置を含んでいてもよい。そのようなヒューマン・インターフェース出力装置は、たとえば、触覚出力、音、光、および臭い/味を通じて、一または複数の人間ユーザーの感覚を刺激するものであってもよい。そのようなヒューマン・インターフェース出力装置は、触覚出力装置(たとえば、タッチスクリーン(1010)、データグローブ(図示せず)、またはジョイスティック(1005)による触覚フィードバック;ただし、入力装置のはたらしをしない触覚フィードバック装置もありうる)、音声出力装置(たとえば、スピーカー(1009)、ヘッドフォン(図示せず))、視覚出力装置(たとえば、CRT画面、LCD画面、プラズマスクリーン、OLED画面を含む画面(1010);それぞれはタッチスクリーン入力機能があってもなくてもよく、それぞれは触覚フィードバック機能があってもなくてもよく、そのうちのいくつかは、2次元の視覚出力または立体視出力のような手段を通じた3次元より高い出力を出力することができる;仮想現実感眼鏡(図示せず)、ホログラフィーディスプレイおよび煙タンク(図示せず))、およびプリンタ(図示せず)を含んでいてもよい。これらの出力装置(画面(1010)など)は、グラフィック・アダプター(1050)を通じてシステム・バス(1048)に接続されることができる。

【0212】

コンピュータ・システム(1000)はまた、人間がアクセス可能な記憶装置および関連する媒体、たとえば、CD/DVDまたは類似の媒体(1021)とともにCD/DVD ROM/RW(1020)を含む光学式媒体、サムドライブ(1022)、取り外し可能なハードドライブまたはソリッドステートドライブ(1023)、テープおよびフロッピーディスクといったレガシー磁気媒体(図示せず)、セキュリティ・ドングルのような特化したROM/ASIC/PLDベースの装置(図示せず)などを含むことができる。

【0213】

当業者はまた、現在開示されている主題に関連して使用される用語「コンピュータ読み取り可能な媒体」は、伝送媒体、搬送波、または他の一時的な信号を包含しないことを理解すべきである。

【0214】

コンピュータ・システム(1000)はまた、一つまたは複数の通信ネットワーク(1055)へのネットワーク・インターフェース(1054)を含むことができる。該一つまたは複数の通信ネットワーク(1055)は、たとえば、無線、有線、光学式でありうる。該一つまたは複数の通信ネットワーク(1055)は、さらに、ローカル、広域、都市圏、車載および工業用、リアルタイム、遅延耐性などでありうる。該一つまたは複数の通信ネットワーク(1055)の例は、イーサネット〔登録商標〕、無線LAN、GSM、3G、4G、5G、LTEなどを含むセルラー・ネットワーク、ケーブルテレビ、衛星テレビ、地上放送テレビを含むTV有線または無線の広域デジタルネットワーク、CANBusを含む車載および工業用などを含む。ある種のネットワークは、普通、ある種の汎用データ・ポートまたは周辺バス(1049)(たとえば、コンピュータ・システム(1000)のUSBポートなど)に取り付けられる外部ネットワーク・インターフェース・アダプターを必要とする。他は、普通、後述するようなシステム・バスへの取り付けによって、コンピュータ・システム(1000)のコアに統合される(たとえば、PCコンピュータ・システムへのイーサネット・インターフェースまたはスマートフォン・コンピュータ・システムへのセルラー・ネットワーク・インターフェース)。これらのネットワークのいずれかを使用して、コンピュータ・シス

テム(1000)は、他のエンティティと通信することができる。そのような通信は、一方向性、受信のみ(たとえば、放送テレビ)、一方向性送信専用(たとえば、ある種のCANbus装置へのCANbus)、または、たとえば、ローカルまたは広域デジタルネットワークを使用する他のコンピュータ・システムへの双方向性であってもよい。上述のようなそれらのネットワークおよびネットワークインターフェースのそれぞれで、ある種のプロトコルおよびプロトコルスタックが使用できる。

【0215】

前述のヒューマン・インターフェース装置、人間がアクセス可能な記憶装置、およびネットワークインターフェースは、コンピュータ・システム(1000)のコア(1040)に取り付けることができる。

【0216】

コア(1040)は、一つまたは複数の中央処理装置(CPU)(1041)、グラフィックス処理装置(GPU)(1042)、フィールドプログラマブルゲートアレイ(FPGA)(1043)の形の特化したプログラマブル処理装置、ある種のタスクのためのハードウェアアクセラレータ(1044)などを含むことができる。これらの装置は、読み出し専用メモリ(ROM)(1045)、ランダムアクセスメモリ(1046)、内部のユーザー・アクセス可能でないハードドライブ、ソリッドステートドライブ(SSD)などの内部大容量記憶装置(1047)とともに、システム・バス(1048)を通じて接続されうる。いくつかのコンピュータ・システムでは、追加のCPU、GPUなどによる拡張を可能にするために、システム・バス(1048)は、一つまたは複数の物理プラグの形でアクセス可能であってもよい。周辺装置は、コアのシステム・バス(1048)に直接取り付けられることも、周辺バス(1049)を通じて取り付けられることもできる。周辺バスのためのアーキテクチャーは、PCI、USBなどを含む。

【0217】

CPU(1041)、GPU(1042)、FPGA(1043)、およびアクセラレータ(1044)は、組み合わせて上述のコンピュータコードを構成することができるある種の命令を、実行することができる。そのコンピュータコードは、ROM(1045)またはRAM(1046)に記憶できる。一時的データも、RAM(1046)に記憶されることができ、一方、持続的データは、たとえば、内部大容量記憶装置(1047)に記憶されることができ。一つまたは複数のCPU(1041)、GPU(1042)、大容量記憶装置(1047)、ROM(1045)、RAM(1046)などと密接に関連付けることができるキャッシュメモリを使用することを通じて、メモリデバイスのいずれかへの高速な記憶および取り出しを可能にすることができる。

【0218】

コンピュータ読み取り可能な媒体は、さまざまなコンピュータ実装された動作を実行するためのコンピュータコードをその上に有することができる。媒体およびコンピュータコードは、本開示の目的のために特別に設計および構築されたものであってもよく、または、コンピュータ・ソフトウェア分野の技術を有する者に周知であり利用可能な種類のものであってもよい。

【0219】

限定ではなく一例として、アーキテクチャー(1000)、具体的にはコア(1040)を有するコンピュータ・システムは、プロセッサ(CPU、GPU、FPGA、アクセラレータ等を含む)が一つまたは複数の有形のコンピュータ可読媒体に具現化されたソフトウェアを実行することの結果として、機能性を提供することができる。そのようなコンピュータ読み取り可能な媒体は、上記で紹介したようなユーザー・アクセス可能な大容量記憶ならびにコア内部の大容量記憶装置(1047)またはROM(1045)のような非一時的な性質のコア(1040)のある種の記憶に関連する媒体であることができる。本開示のさまざまな実施形態を実装するソフトウェアは、そのような装置に記憶され、コア(1040)によって実行されることができる。コンピュータ読み取り可能な媒体は、特定のニーズに応じて、一つまたは複数のメモリデバイスまたはチップを含むことができる。ソフトウェアは、RAM(

10

20

30

40

50

1046) に記憶されたデータ構造を定義し、ソフトウェアによって定義されたプロセスに従ってそのようなデータ構造を修正することを含む、本明細書に記載された特定のプロセスまたは特定の特定部分を、コア (1040) および具体的にはその中のプロセッサ (CPU、GPU、FPGAなどを含む) に実行させることができる。追加的または代替的に、コンピュータ・システムは、回路 (たとえば、アクセラレータ (1044)) 内に配線された、または他の仕方で具現された論理の結果として機能性を提供することができ、これは、本明細書に記載される特定のプロセスまたは特定のプロセスの特定部分を実行するためのソフトウェアの代わりに、またはそれと一緒に動作することができる。ソフトウェアへの言及は、論理を含み、適宜その逆も可能である。コンピュータ読み取り可能媒体への言及は、適宜、実行のためのソフトウェアを記憶する回路 (たとえば集積回路 (IC))、実行のための論理を具現する回路、またはその両方を包含することができる。本開示は、ハードウェアおよびソフトウェアの任意の好適な組み合わせを包含する。

10

【0220】

本開示は、いくつかの例示的实施形態を記載してきたが、変更、置換、およびさまざまな代替等価物があり、それらは本開示の範囲内にはいる。よって、当業者は、本明細書に明示的に示されていないかまたは記載されていないが、本開示の原理を具現し、よって、本開示の精神および範囲内にある多くのシステムおよび方法を考案することができることが理解されるであろう。

【0221】

付録A：頭字語

20

AMVP: Advanced Motion Vector Prediction (先進動きベクトル予測)

ASIC: Application-Specific Integrated Circuit (特定用途向け集積回路)

ATMVP: Alternative/Advanced Temporal Motion Vector Prediction (代替/先進時間的動きベクトル予測)

BMS: Benchmark Set (ベンチマークセット)

BV: Block Vector (ブロックベクトル)

CANBus: Controller Area Network Bus (コントローラエリアネットワークバス)

CB: Coding Block (符号化ブロック)

CD: Compact Disc (コンパクトディスク)

CPR: Current Picture Referencing (現在ピクチャー参照)

30

CPU: Central Processing Unit (中央処理装置)

CRT: Cathode Ray Tube (陰極線管)

CTB: Coding Tree Block (符号化ツリーブロック)

CTU: Coding Tree Unit (符号化ツリー単位)

CU: Coding Unit (符号化単位)

DPB: Decoder Picture Buffer (デコーダ・ピクチャー・バッファ)

DVD: Digital Video Disc (デジタルビデオディスク)

FPGA: Field Programmable Gate Area (フィールド・プログラマブル・ゲートエリア)

GOP: Group of Pictures (ピクチャーグループ)

40

GPU: Graphics Processing Unit (グラフィックス処理ユニット)

GSM: Global System for Mobile communications (グローバル移動通信システム)

HEVC: High Efficiency Video Coding (高効率ビデオ符号化)

HRD: Hypothetical Reference Decoder (仮説参照デコーダ)

IBC: Intra Block Copy (イントラブロックコピー)

IC: Integrated Circuit (集積回路)

JEM: Joint Exploration Model (共同探索モデル)

JVET: Joint Video Exploration Team (共同ビデオ探索チーム)

LAN: Local Area Network (ローカルエリアネットワーク)

LCD: Liquid-Crystal Display (液晶ディスプレイ)

50

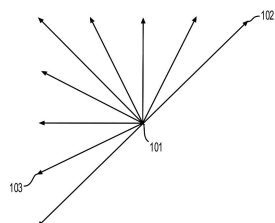
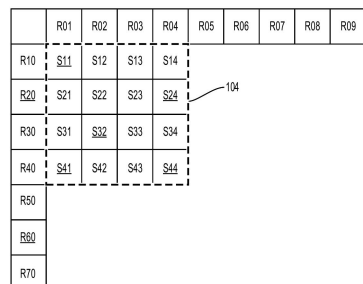
LTE : Long-Term Evolution (ロングタームエボリューション)
 MV : Motion Vector (動きベクトル)
 OLED : Organic Light-Emitting Diode (有機発光ダイオード)
 PB : Prediction Block (予測ブロック)
 PCI : Peripheral Component Interconnect (周辺コンポーネント相互接続)
 PH : Picture Header (ピクチャーヘッダ)
 PLD : Programmable Logic Device (プログラマブルロジックデバイス)
 PPS : Picture Parameter Setting (ピクチャーパラメータセッティング)
 POC : Picture Order Count (ピクチャー順カウント)
 PU : Prediction Unit (予測単位)
 RAM : Random Access Memory (ランダムアクセスメモリ)
 RBSP : Raw Byte Sequence Payload (生のバイト・シーケンス・ペイロード)
 ROM : Read-Only Memory (読み出し専用メモリ)
 RPL : Reference Picture List (参照ピクチャーリスト)
 SCC : Screen Content Coding (スクリーンコンテンツ符号化)
 SDR : Standard Dynamic Range (標準ダイナミックレンジ)
 SEI : Supplementary Enhancement Information (補足向上情報)
 SNR : Signal Noise Ratio (信号対雑音比)
 SPS : Sequence Parameter Set (シーケンスパラメータセット)
 SSD : Solid-state Drive (ソリッドステートドライブ)
 TU : Transform Unit (変換単位)
 USB : Universal Serial Bus (ユニバーサルシリアルバス)
 VUI : Video Usability Information (ビデオユーザビリティ情報)
 VVC : Versatile Video Coding (多用途ビデオ符号化)

10

20

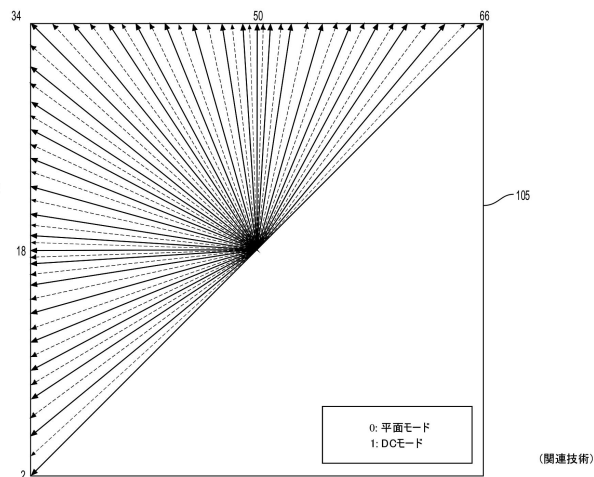
【図面】

【図 1 A】



(関連技術)

【図 1 B】



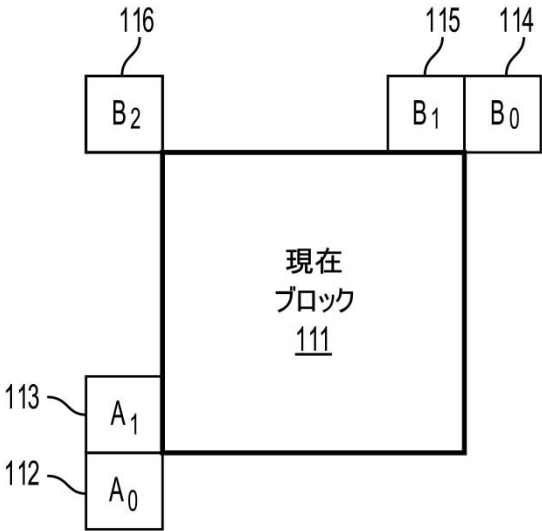
(関連技術)

30

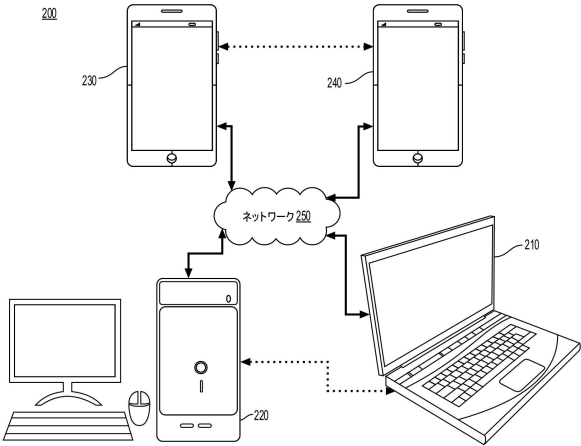
40

50

【図 1 C】



【図 2】

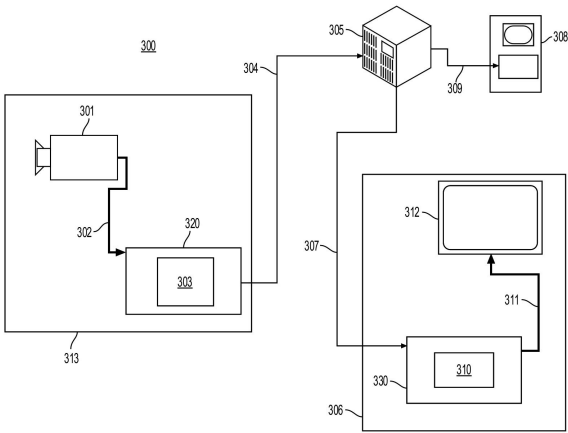


10

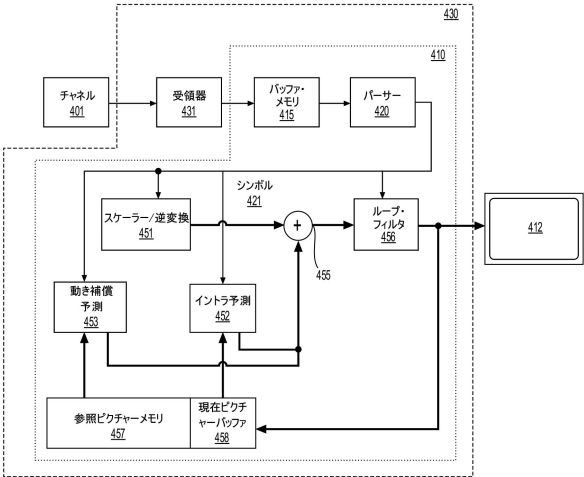
20

(関連技術)

【図 3】



【図 4】

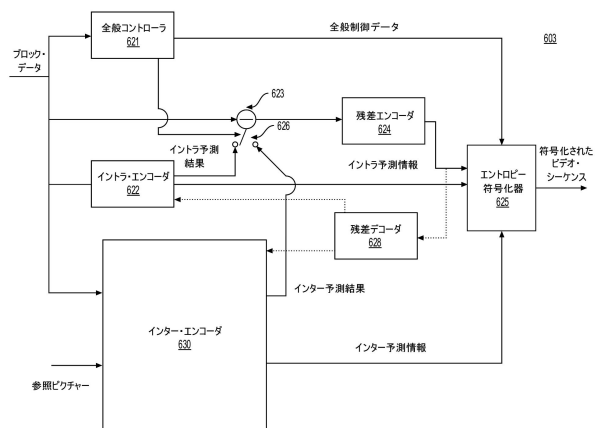


30

40

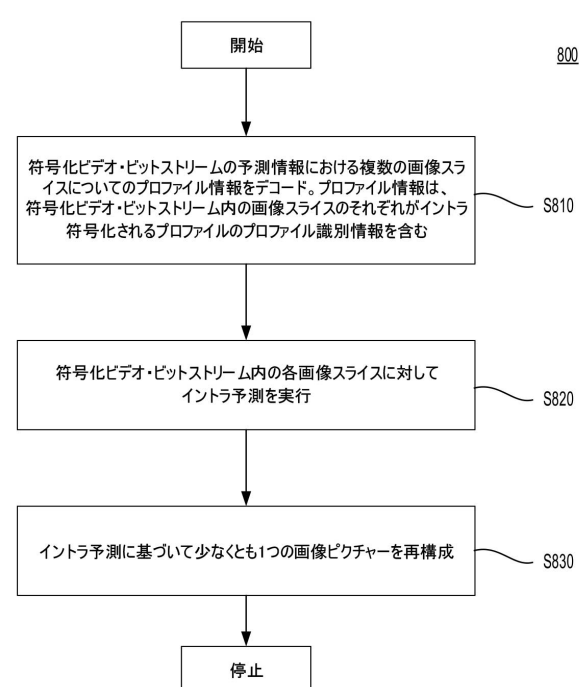
50

【 図 6 】



10

【圖 8】



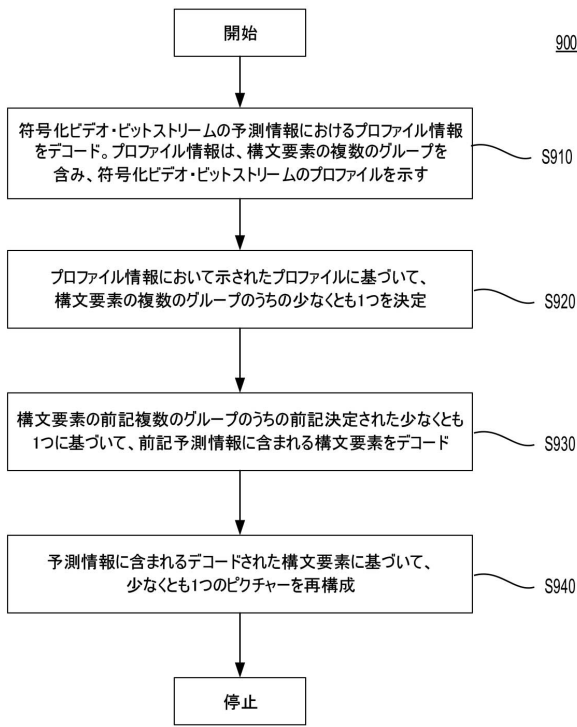
20

30

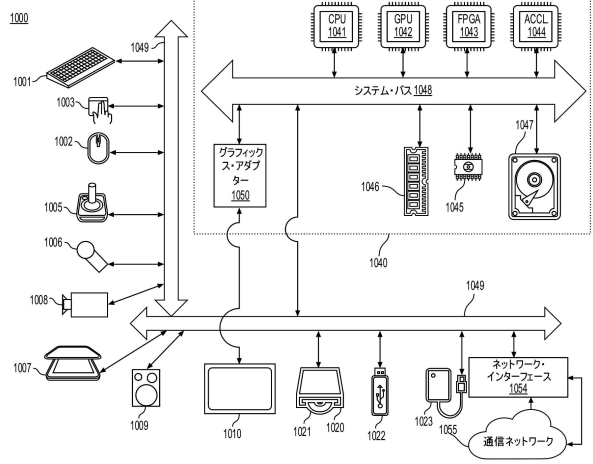
40

50

【図 9】



【図 10】



フロントページの続き

(33)優先権主張国・地域又は機関

米国(US)

(31)優先権主張番号 17/220,481

(32)優先日 令和3年4月1日(2021.4.1)

(33)優先権主張国・地域又は機関

米国(US)

エルシー

(72)発明者 チョイ, ビョンドゥ

アメリカ合衆国 9 4 3 0 6 カリフォルニア州 パロアルト パーク・ブールバード 2 7 4 7 テンセント アメリカ エルエルシー

(72)発明者 リ, シアン

アメリカ合衆国 9 4 3 0 6 カリフォルニア州 パロアルト パーク・ブールバード 2 7 4 7 テンセント アメリカ エルエルシー

(72)発明者 ウェンジャー, ステファン

アメリカ合衆国 9 4 3 0 6 カリフォルニア州 パロアルト パーク・ブールバード 2 7 4 7 テンセント アメリカ エルエルシー

(72)発明者 リウ, シャン

アメリカ合衆国 9 4 3 0 6 カリフォルニア州 パロアルト パーク・ブールバード 2 7 4 7 テンセント アメリカ エルエルシー

審査官 岩井 健二

(56)参考文献 国際公開第 2 0 1 5 / 0 9 8 5 6 1 (WO , A 1)

Jonatan Samuelsson, Sachin Deshpande, and Andrew Segall , AHG17: On Profile, tier, and level syntax structure , Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 , JVET-P0217 , 16th Meeting: Geneva, CH , 2019年09月 , pp.1-2
Benjamin Bross, Jianle Chen, Shan Liu, and Ye-Kui Wang , Versatile Video Coding (Draft 8) , Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 , JVET-Q2001 (version 15) , 17th Meeting: Brussels, BE , 2020年03月12日 , pp.47-50,55-57,119-126,133-137

(58)調査した分野 (Int.Cl. , D B 名)

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