

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2016-174385

(P2016-174385A)

(43) 公開日 平成28年9月29日(2016.9.29)

(51) Int.Cl.	F I	テーマコード (参考)
HO4N 19/70 (2014.01)	HO4N 19/70	5C159
HO4N 19/30 (2014.01)	HO4N 19/30	
HO4N 19/503 (2014.01)	HO4N 19/503	

審査請求 有 請求項の数 4 O L (全 41 頁)

(21) 出願番号	特願2016-92752 (P2016-92752)	(71) 出願人	596099882
(22) 出願日	平成28年5月2日 (2016.5.2)		エレクトロニクス アンド テレコミュニ ケーションズ リサーチ インスティテュー ト
(62) 分割の表示	特願2015-506892 (P2015-506892) の分割		ELECTRONICS AND TEL ECOMMUNICATIONS RES EARCH INSTITUTE
原出願日	平成25年4月16日 (2013.4.16)		大韓民国 305-700 デジョン ユ ソン-グ ガジョン-ロ 218
(31) 優先権主張番号	10-2012-0038870	(74) 代理人	100091982
(32) 優先日	平成24年4月16日 (2012.4.16)		弁理士 永井 浩之
(33) 優先権主張国	韓国 (KR)	(74) 代理人	100091487
(31) 優先権主張番号	10-2012-0066606		弁理士 中村 行孝
(32) 優先日	平成24年6月21日 (2012.6.21)	(74) 代理人	100082991
(33) 優先権主張国	韓国 (KR)		弁理士 佐藤 泰和
(31) 優先権主張番号	10-2012-0067925		
(32) 優先日	平成24年6月25日 (2012.6.25)		
(33) 優先権主張国	韓国 (KR)		

最終頁に続く

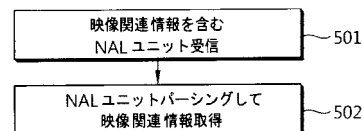
(54) 【発明の名称】 映像情報デコーディング方法、映像デコーディング方法及びそれを利用する装置

(57) 【要約】 (修正有)

【課題】階層的ビットストリーム内の抽出及びスケラ  
ピリティ情報を表現する方法及び装置を提供する。

【解決手段】映像情報デコーディング方法は、エンコー  
ディングされた映像に関連した情報を含むNAL (Ne  
twork Abstraction Layer)ユニ  
ットを含むビットストリームを受信するステップと、前  
記NALユニットのNALユニットヘッダをパースング  
するステップとを含む。前記NALユニットヘッダは、  
エンコーディング時、全体ビットストリームで非参照ピ  
クチャであるか、または参照ピクチャであるかを示す1  
ビットのフラグ情報を含まない。

【選択図】 図5



## 【特許請求の範囲】

## 【請求項 1】

エンコードされた映像に関連した情報を含むNAL(Network Abstraction Layer)ユニットのNALユニットヘッダを受信し、前記NALユニットのNALユニットヘッダをパーシングするパーシング部;を含み

前記NALユニットがビットストリームで非参照ピクチャであるか、または参照ピクチャであるかを示す情報を含まないことを特徴とする映像デコーディング装置。

## 【請求項 2】

前記NALユニットヘッダは、スケーラブル階層をサポートするビットストリームで前記スケーラブル階層を識別するための階層識別情報を含むことを特徴とする請求項1に記載の映像デコーディング装置。

10

## 【請求項 3】

エンコードされた映像に関連した情報を含むNAL(Network Abstraction Layer)ユニットのNALユニットヘッダを受信する段階と、

前記NALユニットのNALユニットヘッダをパーシングする段階;を含み

前記NALユニットがビットストリームで非参照ピクチャであるか、または参照ピクチャであるかを示す情報を含まないことを特徴とする映像デコーディング方法。

## 【請求項 4】

前記NALユニットヘッダは、スケーラブル階層をサポートするビットストリームで前記スケーラブル階層を識別するための階層識別情報を含むことを特徴とする請求項3に記載の映像デコーディング方法。

20

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

本発明は、ビデオ符号化及び復号化処理に関し、より詳しくは、ビットストリーム内のビデオの情報をデコーディングする方法及び装置に関する。

## 【背景技術】

## 【0002】

最近、HD(High Definition)解像度を有する放送サービスが韓国内だけではなく、世界的に拡大されるにつれて、多くのユーザが高解像度、高画質の映像に慣れており、それによって、多くの機関が次世代映像機器に対する開発に拍車を掛けている。また、HDTVと共にHDTVの4倍以上の解像度を有するUHD(Ultra High Definition)に対する関心が増大しながら、より高い解像度、高画質の映像に対する圧縮技術が要求されている。

30

## 【0003】

映像圧縮のために、時間的に以前及び/または以後のピクチャから現在ピクチャに含まれている画素値を予測するインター(inter)予測技術、現在ピクチャ内の画素情報を利用して現在ピクチャに含まれている画素値を予測するイントラ(intra)予測技術、出現頻度が高いシンボル(symbol)に短い符号を割り当て、出現頻度が低いシンボルに長い符号を割り当てるエントロピー符号化技術などが使われることができる。

40

## 【0004】

映像圧縮技術には、流動的なネットワーク環境を考慮せずにハードウェアの制限的な動作環境下で一定のネットワーク帯域幅を提供する技術がある。しかし、随時帯域幅が変化するネットワーク環境に適用される映像データを圧縮するためには新たな圧縮技術が要求され、そのために、スケーラブル(scalable)ビデオ符号化/復号化方法が使われることができる。

## 【発明の概要】

## 【発明が解決しようとする課題】

## 【0005】

本発明の技術的課題は、階層的ビットストリーム内の抽出及びスケーラビリティ情報を

50

記述する方法及び装置を提供することである。

【0006】

本発明の他の技術的課題は、柔軟な方式で多様な種類のビットストリームのスケラビリティ情報を表現する方法及び装置を提供することである。

【0007】

本発明の他の技術的課題は、階層的ビットストリーム内の抽出及びスケラビリティ情報をパケットレベルでの適応的に変換可能に提供する方法及び装置を提供することである。

【課題を解決するための手段】

【0008】

本発明の一実施例に係る映像情報デコーディング方法は、エンコーディングされた映像に関連した情報を含むNAL(Network Abstraction Layer)ユニットを含むビットストリームを受信するステップと、前記NALユニットのNALユニットヘッダをパッシングするステップとを含み、前記NALユニットヘッダは、前記NALユニットのエンコーディング時、全体ビットストリームで非参照ピクチャであるか、または参照ピクチャであるかを示す1ビットのフラグ情報を含まない。

【0009】

本発明の一実施例に係る映像デコーディング方法は、受信されたピクチャをデコーディングするステップと、デコーディングされたピクチャをDPB(decoded picture buffer)に参照ピクチャで表示するステップと、前記デコーディングされたピクチャの次のピクチャに対するスライスヘッダをパッシングするステップと、前記スライスヘッダに含まれている参照ピクチャ情報に基づき、前記デコーディングされたピクチャが参照ピクチャであるか、または非参照ピクチャであるかを表示するステップとを含む。

【0010】

本発明の他の実施例として、活性化されるパラメータセットに対する情報を含むSEI(Supplemental enhancement information)メッセージを受信し、前記パラメータセットに対する情報をパッシングする。

【発明の効果】

【0011】

本発明の一実施例によると、階層的ビットストリーム内の抽出及びスケラビリティ情報を記述する方法及び装置が提供される。

【0012】

本発明の一実施例によると、柔軟な方式で多様な種類のビットストリームのスケラビリティ情報を表現する方法及び装置が提供される。

【0013】

本発明の他の実施例によると、階層的ビットストリーム内の抽出及びスケラビリティ情報をパケットレベルでの適応的に変換可能に提供する方法及び装置が提供される。

【図面の簡単な説明】

【0014】

【図1】映像符号化装置の一実施例に係る構成を示すブロック図である。

【図2】映像復号化装置の一実施例に係る構成を示すブロック図である。

【図3】本発明が適用されることができ、複数階層を利用したスケラブルビデオコーディング構造の一実施例を概念的に示す概念図である。

【図4】本発明による映像情報のエンコーディング方法を説明するための制御流れ図である。

【図5】本発明による映像情報のデコーディング方法を説明するための制御流れ図である。

【発明を実施するための形態】

【0015】

10

20

30

40

50

以下、図面を参照して本発明の実施形態に対して具体的に説明する。本明細書の実施例を説明するにあたって、関連した公知構成または機能に対する具体的な説明が本明細書の要旨を不明にすると判断される場合、その詳細な説明は省略する。

【0016】

一構成要素が他の構成要素に“連結されている”または“接続されている”と言及された場合、該当他の構成要素に直接的に連結されている、または接続されていることもあるが、中間に他の構成要素が存在することもあると理解しなければならない。また、本発明において、特定構成を“含む”と記述する内容は、該当構成以外の構成を排除するものではなく、追加的な構成が本発明の実施または本発明の技術的思想の範囲に含まれることができることを意味する。

10

【0017】

第1、第2などの用語は、多様な構成要素の説明に使われることができるが、前記構成要素は、前記用語により限定されてはならない。前記用語は、一つの構成要素を他の構成要素から区別する目的にのみ使われる。例えば、本発明の権利範囲を外れない限り、第1の構成要素は第2の構成要素と命名することができ、同様に、第2の構成要素も第1の構成要素と命名することができる。

【0018】

また、本発明の実施例に示す構成部は、互いに異なる特徴的な機能を示すために独立的に図示されるものであり、各構成部が分離されたハードウェアや一つのソフトウェア構成単位に構成されることを意味しない。即ち、各構成部は、説明の便宜上、各々の構成部として羅列して含むものであり、各構成部のうち少なくとも2個の構成部が統合されて一つの構成部からなり、または一つの構成部が複数の構成部に分けられて機能を遂行することができ、このような各構成部の統合された実施例及び分離された実施例も本発明の本質から外れない限り、本発明の権利範囲に含まれる。

20

【0019】

また、一部の構成要素は、本発明で本質的な機能を遂行する必須な構成要素ではなく、単に性能を向上させるための選択的構成要素である。本発明は、単に性能向上のために使われる構成要素を除いた本発明の本質を具現するのに必須な構成部のみを含んで具現されることができ、単に性能向上のために使われる選択的構成要素を除いた必須構成要素のみを含む構造も本発明の権利範囲に含まれる。

30

【0020】

図1は、映像符号化装置の一実施例に係る構成を示すブロック図である。スケーラブル(scalable)ビデオ符号化/復号化方法または装置は、スケーラビリティ(scalability)を提供しない一般的な映像符号化/復号化方法または装置の拡張(extension)により具現されることができ、図1のブロック図は、スケーラブルビデオ符号化装置のベースになることができる映像符号化装置の一実施例を示す。

【0021】

図1を参照すると、前記映像符号化装置100は、動き予測部111、動き補償部112、イントラ予測部120、スイッチ115、減算器125、変換部130、量子化部140、エントロピー符号化部150、逆量子化部160、逆変換部170、加算器175、フィルタ部180及び参照映像バッファ190を含む。

40

【0022】

映像符号化装置100は、入力映像に対してイントラ(intra)モードまたはインター(inter)モードに符号化を実行することで、ビットストリーム(bit stream)を出力することができる。イントラ予測は画面内予測を意味し、インター予測は画面間予測を意味する。イントラモードである場合、スイッチ115がイントラに切り替えられ、インターモードである場合、スイッチ115がインターに切り替えられる。映像符号化装置100は、入力映像の入力ブロックに対する予測ブロックを生成した後、入力ブロックと予測ブロックとの差分を符号化することができる。

【0023】

50

イントラモードである場合、イントラ予測部 120 は、現在ブロック周辺の既に符号化されたブロックの画素値を利用して空間的予測を実行することで、予測ブロックを生成することができる。

【0024】

インターモードである場合、動き予測部 111 は、動き予測過程で参照映像バッファ 190 に格納されている参照映像で入力ブロックと最もよくマッチされる領域を探して動きベクトルを求めることができる。動き補償部 112 は、動きベクトルと参照映像バッファ 190 に格納されている参照映像を利用して動き補償を実行することによって、予測ブロックを生成することができる。

【0025】

減算器 125 は、入力ブロックと生成された予測ブロックとの差分により残余ブロック (residual block) を生成することができる。変換部 130 は、残余ブロックに対して変換 (transform) を実行することで、変換係数 (transform coefficient) を出力することができる。そして、量子化部 140 は、入力された変換係数を量子化パラメータによって量子化し、量子化された係数 (quantized coefficient) を出力することができる。

【0026】

エントロピー符号化部 150 は、量子化部 140 で算出された値または符号化過程で算出された符号化パラメータ値などに基き、シンボル (symbol) を確率分布によってエントロピー符号化することで、ビットストリーム (bit stream) を出力することができる。エントロピー符号化方法は、多様な値を有するシンボルの入力を受け、統計的な重複性を除去すると共に、復号可能な 2 進数の列で表現する方法である。

【0027】

ここで、シンボルとは、符号化/復号化対象構文要素 (syntax element) 及び符号化パラメータ (coding parameter)、残余信号 (residual signal) の値などを意味する。符号化パラメータは、符号化及び復号化に必要な媒介変数であって、構文要素のように符号化器で符号化されて復号化器に伝達される情報だけでなく、符号化または復号化過程で類推されることができる情報を含むことができ、映像を符号化または復号化する時に必要な情報を意味する。符号化パラメータは、例えば、イントラ/インター予測モード、移動/動きベクトル、参照映像索引、符号化ブロックパターン、残余信号有無、変換係数、量子化された変換係数、量子化パラメータ、ブロックの大きさ、ブロック分割情報などの値または統計を含むことができる。また、残余信号は、原信号と予測信号との差を意味し、また、原信号と予測信号との差が変換 (transform) された形態の信号または原信号と予測信号との差が変換されて量子化された形態の信号を意味する。残余信号は、ブロック単位では残余ブロックという。

【0028】

エントロピー符号化が適用される場合、高い発生確率を有するシンボルに少ない数のビットが割り当てられ、低い発生確率を有するシンボルに多い数のビットが割り当てられてシンボルが表現されることによって、符号化対象シンボルに対するビット列の大きさが減少されることができる。したがって、エントロピー符号化を介して映像符号化の圧縮性能が高まることができる。

【0029】

エントロピー符号化のために、指数ゴロム (exponential golomb)、CAVLC (Context-Adaptive Variable Length Coding)、CABAC (Context-Adaptive Binary Arithmetic Coding) のような符号化方法が使われることができる。例えば、エントロピー符号化部 150 には、可変長さ符号化 (VLC: Variable Length Coding/Code) テーブルのようなエントロピー符号化を実行するためのテーブルが格納されることができ、エントロピー符号化部 150 は、格納された可変長さ符号化 (VLC) テーブルを使用してエントロピー符号化を実行することができる。また、エントロ

10

20

30

40

50

ピー符号化部 150 は、対象シンボルの 2 進化 (binarization) 方法及び対象シンボル/ピン (bin) の確率モデル (probability model) を導出した後、導出された 2 進化方法または確率モデルを使用してエントロピー符号化を実行することもできる。

【0030】

量子化された係数は、逆量子化部 160 で逆量子化され、逆変換部 170 で逆変換されることができる。逆量子化、逆変換された係数は、加算器 175 を介して予測ブロックと加えられて復元ブロックが生成されることができる。

【0031】

復元ブロックは、フィルタ部 180 を経て、フィルタ部 180 は、デブロッキングフィルタ (deblocking filter)、SAO (Sample Adaptive Offset)、ALF (Adaptive Loop Filter) のうち少なくとも一つ以上を復元ブロックまたは復元ピクチャに適用することができる。フィルタ部 180 を経た復元ブロックは、参照映像バッファ 190 に格納されることができる。

10

【0032】

図 2 は、映像復号化装置の一実施例に係る構成を示すブロック図である。図 1 で詳述したように、スケーラブルビデオ符号化/復号化方法または装置は、スケーラビリティを提供しない一般的な映像符号化/復号化方法または装置の拡張により具現されることができ、図 2 のブロック図は、スケーラブルビデオ復号化装置のベースになることができる映像復号化装置の一実施例を示す。

20

【0033】

図 2 を参照すると、前記映像復号化装置 200 は、エントロピー復号化部 210、逆量子化部 220、逆変換部 230、イントラ予測部 240、動き補償部 250、フィルタ部 260 及び参照映像バッファ 270 を含む。

【0034】

映像復号化装置 200 は、符号化器から出力されたビットストリームの入力を受けてイントラモードまたはインターモードに復号化を実行することで、再構成された映像、即ち、復元映像を出力することができる。イントラモードである場合、スイッチがイントラに切り替えられ、インターモードである場合、スイッチがインターに切り替えられる。映像復号化装置 200 は、入力されたビットストリームから復元された残余ブロック (residual block) を得て、予測ブロックを生成した後、復元された残余ブロックと予測ブロックを加えて再構成されたブロック、即ち、復元ブロックを生成することができる。

30

【0035】

エントロピー復号化部 210 は、入力されたビットストリームを確率分布によってエントロピー復号化し、量子化された係数 (quantized coefficient) 形態のシンボルを含むシンボルを生成することができる。エントロピー復号化方法は、2 進数の列の入力を受けて各シンボルを生成する方法である。エントロピー復号化方法は、前述したエントロピー符号化方法と同様である。

【0036】

量子化された係数は、逆量子化部 220 で逆量子化され、逆変換部 230 で逆変換され、量子化された係数が逆量子化/逆変換された結果、復元された残余ブロック (residual block) が生成されることができる。

40

【0037】

イントラモードである場合、イントラ予測部 240 は、現在ブロック周辺の既に符号化されたブロックの画素値を利用して空間的予測を実行することで、予測ブロックを生成することができる。インターモードである場合、動き補償部 250 は、動きベクトル及び参照映像バッファ 270 に格納されている参照映像を利用して動き補償を実行することによって、予測ブロックを生成することができる。

【0038】

50

復元された残余ブロックと予測ブロックは、加算器 255 を介して加えられ、加えられたブロックは、フィルタ部 260 を経る。フィルタ部 260 は、デブロッキングフィルタ、SAO、ALF のうち少なくとも一つ以上を復元ブロックまたは復元ピクチャに適用することができる。フィルタ部 260 は、再構成された映像、即ち、復元映像を出力する。復元映像は、参照映像バッファ 270 に格納されて画面間予測に使われることができる。

#### 【0039】

前記映像復号化装置 200 に含まれているエントロピー復号化部 210、逆量子化部 220、逆変換部 230、イントラ予測部 240、動き補償部 250、フィルタ部 260 及び参照映像バッファ 270 のうち、映像の復号化に直接的に関連した構成要素、例えば、エントロピー復号化部 210、逆量子化部 220、逆変換部 230、イントラ予測部 240、動き補償部 250、フィルタ部 260 などを他の構成要素と区分して復号化部またはデコーディング部で表現することができる。

10

#### 【0040】

また、映像復号化装置 200 は、ビットストリームに含まれているエンコーディングされた映像に関連した情報をパーシングするパーシング部(図示せず)をさらに含むことができる。パーシング部は、エントロピー復号化部 210 を含むこともでき、エントロピー復号化部 210 に含まれることもできる。また、このようなパーシング部は、デコーディング部の一つの構成要素で具現されることもできる。

#### 【0041】

図 3 は、本発明が適用されることができる、複数階層を利用したスケーラブルビデオコーディング構造の一実施例を概念的に示す概念図である。図 3 において、GOP(Group of Picture)は、ピクチャ群、即ち、ピクチャのグループを示す。

20

#### 【0042】

映像データを送信するためには送信媒体が必要であり、その性能は、多様なネットワーク環境によって送信媒体別に異なる。このような多様な送信媒体またはネットワーク環境への適用のためにスケーラブルビデオコーディング方法が提供されることができる。

#### 【0043】

スケーラブルビデオコーディング方法は、階層(layer)間のテクスチャ情報、動き情報、残余信号などを活用して階層間重複性を除去して符号化/復号化性能を高めるコーディング方法である。スケーラブルビデオコーディング方法は、送信ビット率、送信エラー率、システムリソースなどの周辺条件によって、空間的、時間的、画質的な観点で多様なスケーラビリティを提供することができる。

30

#### 【0044】

スケーラブルビデオコーディングは、多様なネットワーク状況に適用可能なビットストリームを提供できるように、複数階層(multiple layers)構造を使用して実行されることができる。例えば、スケーラブルビデオコーディング構造は、一般的な映像符号化方法を利用して映像データを圧縮処理する基本階層を含むことができ、基本階層の符号化情報及び一般的な映像符号化方法を共に使用して映像データを圧縮処理する向上階層を含むことができる。

#### 【0045】

ここで、階層(layer)は、空間(spatial、例えば、映像の大きさ)、時間(temporal、例えば、符号化順序、映像出力順序、フレームレート)、画質、複雑度などを基準にした区分される映像及びビットストリーム(bit stream)の集合を意味する。また、基本階層は、参照階層またはBase layerを意味し、向上階層は、Enhancement layerを意味する。また、複数の階層は、相互間に従属性を有することもできる。

40

#### 【0046】

図 3 を参照すると、例えば、基本階層は、SD(standard definition)、15 Hz のフレーム率、1 Mbps ビット率で定義されることができ、第 1 の向上階層は、HD(high definition)、30 Hz のフレーム率、3.9 Mbps

50

ビット率で定義されることができ、第2の向上階層は、4K-UHE(ultra high definition)、60Hzのフレーム率、27.2Mbpsビット率で定義されることができる。前記フォーマット(format)、フレーム率、ビット率等は、一つの実施例に過ぎず、必要によって異なるように決まることができる。また、使われる階層の数も本実施例に限定されるものではなく、状況によって異なるように決まることができる。

【0047】

例えば、送信帯域幅が4Mbpsの場合、前記第1の向上階層HDのフレームレートを減らして15Hz以下に送信することができる。スケーラブルビデオコーディング方法は、前記図3の実施例で詳述した方法により時間的、空間的、画質的スケーラビリティを提供することができる。

10

【0048】

以下、スケーラブルビデオコーディングは、符号化観点において、スケーラブルビデオ符号化と同じ意味を有し、復号化観点において、スケーラブルビデオ復号化と同じ意味を有する。

【0049】

前述したように、異種の通信網及び多様な端末によって、スケーラビリティは、現在ビデオフォーマットの重要な機能になった。AVC(Advanced Video Coding)の拡張標準であるSVC(Scalable Video Coding)は、圧縮効率を最大限維持すると共に、多様な範囲のビットレートを有するビットストリームを生成することができるように開発された。多様なデバイス及びネットワークの特性と変化を満たすために、SVCビットストリームは、容易に多様な方式で抽出されることができる。即ち、SVC標準は、空間的(spatial)、時間的(temporal)、画質(SNR)スケーラビリティを提供する。

20

【0050】

一方、複数の階層を含むビットストリームは、パケットスイッチングネットワーク(packet-switching network)を介してビデオの適応的送信を容易にするNAL(Network Abstraction Layer)unitで構成される。複数の階層と同様に、ビットストリーム内の複数の多視点映像を含むマルチビュービデオコーディング(multi-view video coding)における複数の視点間の関係は、複数のレイヤをサポートするビデオでの空間的階層(spatial layer)間の関係と同様である。

30

【0051】

コンテンツ伝達経路(content delivery path)における全てのノードでビットストリームを効果的且つ効率的に変換するためにはビットストリームのスケーラビリティ情報が相当重要である。現在単一レイヤに対するビデオコーディングの標準(high efficiency video coding)では、NAL Unit(以下、NALU)ヘッダ(header)に階層情報と関連した2個のフィールドであるtemporal\_idとreserved\_one\_5bitsが存在する。3bitsの長さを有するtemporal\_idは、ビデオビットストリームの時間的レイヤ(temporal layer)を示し、reserved\_one\_5bitsは、以後に他の階層情報を示すための領域に該当する。

40

【0052】

時間的レイヤは、ビデオコーディングレイヤ(video coding layer、VCL)NAL unitで構成された時間的にスケールされることができるビットストリームのレイヤを意味し、時間的レイヤは、特定のtemporal\_id値を有する。

【0053】

本発明は、複数のレイヤをサポートするビットストリーム内で映像の抽出情報(extraction)及び階層的情報(scalability information)を効果的に記述し、それをシグナリングするための方法及びそれを具現する装置に関する発明

50

である。

【0054】

本発明では、ビットストリームを2つのタイプ、時間スケーラビリティ(temporal scalability)のみをサポートする基本タイプ(base type)と、時間を含む空間/画質/視点をサポートするスケーラビリティを有することができる拡張タイプ(extended type)とに分けて説明する。

【0055】

ビットストリームの第1のタイプは、単一階層ビデオをサポートするビットストリームに対するものであり、第2のタイプは、HEVCベースの階層的ビデオ符号化で向上階層のためのものである。以下、2つのビットストリームタイプのスケーラビリティ情報を表現するための改善案を提案する。本発明によると、拡張タイプにおいて、5bitのreserved\_one\_5bitsは、スケーラブル階層の識別子を示すlayer\_idとして使われることができる。

10

【0056】

NALUヘッダからnal\_ref\_flag除去

【0057】

nal\_ref\_flagは、非参照ピクチャ(non-reference picture)を示すために使われる。この情報は、非参照ピクチャと参照ピクチャ(reference picture)との間の概略的な優先順位を示すが、送信のためのnal\_ref\_flagの使用は多少制限的である。

20

【0058】

参照ピクチャ(Reference picture)は、デコーディング順序上、後続のピクチャ(subsequent pictures)のデコーディング時、画面間予測のために使われることができるサンプルを含むピクチャを意味する。

【0059】

非参照ピクチャ(non-reference picture)は、デコーディング順序上、後続のピクチャのデコーディング時、画面間予測のために使われないサンプルを含むピクチャを意味する。

【0060】

nal\_ref\_flagは、エンコーディング時、該当NALユニットが全体ビットストリーム上で非参照ピクチャであるか、または参照ピクチャであるかを示す情報を示すフラグである。

30

【0061】

nal\_ref\_flagが1の場合、NALUは、SPS(sequence parameter set)、PPS(picture parameter set)、APS(adaptation parameter set)または参照ピクチャのスライスを含むことを意味し、nal\_ref\_flagが0の場合、NALUは、非参照ピクチャの一部または全部を含むスライスを含むことを意味する。

【0062】

このとき、nal\_ref\_flag値が1であるNALUは、参照ピクチャのスライスを含むことができ、nal\_ref\_flagは、VPS(video parameter set)、SPS(sequence parameter set)、PPS(picture parameter set)のNALUに対して1の値を有する。特定ピクチャのVCL NALUのうち一つが、nal\_ref\_flag値が0の場合、該当ピクチャの全てのVCL NALUに対してnal\_ref\_flagは、0の値を有する。

40

【0063】

一方、全ての非参照ピクチャ(non-reference picture)、特に、大部分最上位時間的レイヤに該当する非参照ピクチャが抽出されると、抽出後に残った全てのピクチャのnal\_ref\_flagは1になる。

【0064】

50

しかし、適応変換された(抽出された)ビットストリームの一部ピクチャ、即ち、残ったビットストリームで最上位時間的レイヤに該当するピクチャは、`nal_ref_flag`が1であるとしても非参照ピクチャになる。

【0065】

即ち、NALUヘッダの他のシンタックス要素(例えば、`temporal_id`)が適応変換(抽出)をサポートするのに一層効果的である。即ち、ビットストリームが含む全体時間的レイヤの個数とNALUヘッダの`temporal_id`値を利用し、所望の時間的レイヤを含むビットストリームを抽出することができる。

【0066】

また、`nal_ref_flag`は、`nal_ref_flag`を含むNALUで構成されたピクチャをデコーディング(復元)した後、DPB(`decoded picture buffer`)のようなメモリに格納する時、該当ピクチャを以後に参照ピクチャとして使用するかどうかを表示する時も使われることができる。`nal_ref_flag`が1の場合、以後に参照ピクチャとして使われると表示し、`nal_ref_flag`が0の場合、以後に参照ピクチャとして使用しないと表示することができる。

10

【0067】

`nal_ref_flag`を介して該当NALUが非参照ピクチャであるか、または参照ピクチャであるかを判断せずに、デコーディングされたピクチャをDPBに格納する時、参照ピクチャで表示することができる。この場合、デコーディングされたピクチャが非参照ピクチャであるが、参照ピクチャで表示されても、デコーディング順序上、該当ピクチャの次のピクチャをデコーディングするにあたって、次のピクチャのヘッダ(`slice header`)に伝達される参照ピクチャリストに該当ピクチャが含まれていないため、問題が発生しない。

20

【0068】

即ち、次のピクチャをデコーディングする時、スライスヘッダに含まれている参照ピクチャリストによって以前にデコーディングされたピクチャが参照ピクチャであるか、または非参照ピクチャであるかが表示される。したがって、`nal_ref_flag`を介してデコーディングされたピクチャが参照ピクチャであるかが判断されずに、参照ピクチャで表示されても、デコーディングされたピクチャを参照ピクチャまたは非参照ピクチャと判断するのに問題が発生しない。

30

【0069】

本発明では、NALUヘッダから`nal_ref_flag`を削除し、または`nal_ref_flag`の意味(`semantics`)を変更することを提案する。`nal_ref_flag`削除と関連した実施例は、下記の通りである。

【0070】

#### 実施例1

【0071】

`nal_ref_flag`を`slice_ref_flag`に変更し、フラグの位置をNALUヘッダからスライスヘッダ(`slice header`)に移す。スライスヘッダのシンタックスは、表1のように修正されることできる。

40

【0072】

【表 1】

slice_header( ) {	Descriptor
<b>slice_ref_flag</b>	u(1)
<b>first_slice_in_pic_flag</b>	u(1)
if( first_slice_in_pic_flag == 0 )	
<b>slice_address</b>	u(v)
...	ue(v)
}	

10

## 【0073】

表1において、`slice_ref_flag`の値が1の場合、スライスが参照ピクチャの一部であることを指示し、0の場合、スライスが非参照ピクチャの一部であることを指示する。

## 【0074】

## 実施例 2

## 【0075】

`nal_ref_flag`を`au_ref_flag`に変更し、フラグの位置をNALUヘッダからアクセスユニットデリミタ(`access unit delimiter`)に移す。アクセスユニットデリミタのシンタックスは、表2の通りである。

20

## 【0076】

## 【表 2】

access_unit_delimiter_rbsp( ) {	Descriptor
<b>pic_type</b>	u(3)
<b>au_ref_flag</b>	u(1)
<b>rbps_trailing_bits( )</b>	
}	

30

## 【0077】

表2において、`au_ref_flag`が1の場合、アクセスユニットが参照ピクチャを含むことを指示し、0の場合、アクセスユニットが非参照ピクチャを含むことを指示する。

## 【0078】

## 実施例 3

## 【0079】

`nal_ref_flag`を他のシンタックスに移動させずに、`nal_ref_flag`をNALUヘッダから削除する。

40

## 【0080】

エンコーディング時、全体ビットストリームで非参照ピクチャであるか、または参照ピクチャであるかを示す1ビットのフラグ情報である`nal_ref_flag`が削除されると、`nal_ref_flag`によって実行されたピクチャが参照ピクチャであるかどうかに対する判断は、他の過程を介して実行されることができる。受信されたピクチャをデコーディングした後、デコーディングされたピクチャをDPB(`decoded picture buffer`)に無条件参照ピクチャで表示する。即ち、デコーディングされたピクチャが参照ピクチャであるかどうかを判断せずに、参照ピクチャで表示されることができる。

## 【0081】

50

その後、デコーディングされたピクチャの次のピクチャに対するスライスヘッダをパースングし、スライスヘッダに含まれている参照ピクチャ情報に基づいてデコーディングされたピクチャが参照ピクチャであるか、または非参照ピクチャであるかを表示することができる。

【0082】

実施例 4

【0083】

NALUヘッダから`nal_ref_flag`を削除し、非参照ピクチャのNALUという情報を示すために`temporal_id`を使用することができる。`temporal_id`は“7”、またはビットストリームに含まれている最大時間的レイヤの個数 - 1 (即ち、`max_temporal_layers_minus_1`)、または“0”を除いた既定された値になることができる。

10

【0084】

実施例 5

【0085】

NALUヘッダから`nal_ref_flag`を削除し、非参照ピクチャのNALUという情報を示すために`reserved_one_5bits`を`priority_id`構成要素として使用することができる。`priority_id`は、該当NALUの優先順位(`priority`)を示す識別子であって、異なる空間、時間及び画質に関係なく優先順位によるビットストリーム抽出機能を提供するために使われる。

20

【0086】

即ち、もし、`temporal_id = Ta`が最上位時間的レイヤの識別子である場合、`temporal_id = Ta`であり、`priority_id = 31` (または、他の特定値)であるNALUを非参照ピクチャのNALUであることを示すために使用する。

【0087】

`nal_ref_flag`をシグナリングするために使われた1bitは、下記のうちいずれか一つとして使われることができる。

【0088】

(1) `nal_unit_type`の指示に使われることができる。`nal_unit_type`は、7bits信号になることができ、NALUタイプの個数は、2倍に増加することができる。

30

【0089】

(2) `temporal_id`の指示に使われることができる。`temporal_id`は、4bits信号になることができ、最大時間的レイヤの数が2倍に増加することができる。

【0090】

(3) `layer_id`を指示するために使われることができる。`layer_id`は、階層的ビットストリームのスケーラブル階層の識別子を意味し、`reserved_one_5bits`シンタックス要素によってシグナリングされるすることができる。スケーラブル階層を識別するために使われた`reserved_one_5bits`の5bitに`nal_ref_flag`のシグナリングのために使用した1bitが追加されることで、`layer_id`は6bit信号になることができる。6bitを使用するようになると、64個のスケーラブル階層を識別することができる。

40

【0091】

(4) `reserved_one_5bits`が`priority`を示すかどうかを知らせる`flag`として使われることができる。

【0092】

(5) `reserved_bit`として使われることができる。

【0093】

もし、`nal_ref_flag`をNALUヘッダから削除しない場合、`nal_ref_`

50

flagの意味は、下記のように修正されることができる。

【0094】

nal\_ref\_flagが0の場合、NALUは、非参照ピクチャのスライスのみを含むことを指示し、nal\_ref\_flagが1の場合、NALUは、参照ピクチャまたは非参照ピクチャのスライスを含むことができることを指示する。

【0095】

ビデオパラメータセットの活性化シグナリング

【0096】

ビデオパラメータセット(video parameter set)は、映像をデコーディングするための最も基本的な情報を含み、既存のSPSに存在した内容を含むことができる。

10

【0097】

ビデオパラメータセットには、時間的スケーラビリティをサポートする時間的レイヤを示すサブレイヤ(sub-layer)に対する情報と、空間的(spatial)、クオリティ的(quality)及び視点的(view)スケーラビリティをサポートする複数の階層に対する情報とを含むことができる。即ち、ビデオパラメータセットは、複数の階層情報、即ち、HEVC extensionのためのシンタックスを含むこともできる。

【0098】

A. ビデオパラメータセット(Video Parameter Set)

【0099】

ビデオパラメータセットに対するシンタックスは、表3の通りである。

20

【0100】

【表 3】

	Descriptor	
video_parameter_set_rbsp( ) {		
<b>video_parameter_set_id</b>	ue(v)	
<b>max_temporal_layers_minus1</b>	u(3)	
<b>priority_id_flag</b>	u(1)	
<b>extension_info_flag</b>	u(1)	
if( extension_info_flag == 0 ) {		
<b>pic_width_in_luma_samples</b>	ue(v)	10
<b>pic_height_in_luma_samples</b>	ue(v)	
<b>pic_cropping_flag</b>	u(1)	
if( pic_cropping_flag ) {		
<b>pic_crop_left_offset</b>	ue(v)	
<b>pic_crop_right_offset</b>	ue(v)	
<b>pic_crop_top_offset</b>	ue(v)	
<b>pic_crop_bottom_offset</b>	ue(v)	
}		
<b>bit_depth_luma_minus8</b>	ue(v)	20
<b>bit_depth_chroma_minus8</b>	ue(v)	
<b>chroma_format_idc</b>	u(2)	
<b>temporal_id_nesting_flag</b>	u(1)	
for ( i = 0; i <= num_temporal_layers_minus1; i++ ) {		
<b>profile_idc[ i ]</b>	u(8)	
<b>reserved_zero_8bits[ i ]</b> /* equal to 0 */	u(8)	
<b>level_idc[ i ]</b>	u(8)	
}		
<b>tiles_or_entropy_coding_sync_idc</b>	u(2)	
if( tiles_or_entropy_coding_sync_idc == 1 ) {		30
<b>num_tile_columns_minus1</b>	ue(v)	
<b>num_tile_rows_minus1</b>	ue(v)	
<b>uniform_spacing_flag</b>	u(1)	
if( !uniform_spacing_flag ) {		
for( i = 0; i < num_tile_columns_minus1; i++ )		
<b>column_width[ i ]</b>	ue(v)	
for( i = 0; i < num_tile_rows_minus1; i++ )		
<b>row_height[ i ]</b>	ue(v)	
}		
<b>loop_filter_across_tiles_enabled_flag</b>	u(1)	40
}		
}		
if( extension_info_flag == 1 ) {		
extension_info()		
}		
<b>rbp_trailing_bits( )</b>		
}		

表3において、大部分のシンタックスは、単一レイヤを含むビットストリームに適用されるSPSシンタックスと同じ意味を有し、追加的な部分は、下記の通りである。

【0102】

- `video_parameter_set_id`は、ビデオパラメータセット(`video parameter set`)の識別子を意味し、SPS(`sequence parameter set`)、SEI(`supplemental enhancement information`)、アクセスユニットデリミタ(`access unit delimiter`)で参照されることができる。

【0103】

- `priority_id_flag`が1の場合、`reserved_one_5bits`がSVC標準の`priority_id`と同じく使われることを意味し、`priority_id_flag`が0の場合、`reserved_one_5bits`が`layer_id`として使われることを意味する。

10

【0104】

- `extension_info_flag`が0の場合、ビットストリームがHEVCの単一階層標準に従うことを指示し、1の場合、スケーラビリティのサポートのための向上階層(HEVC extensionをサポートする場合)を示し、階層と関連した情報が提供される。

【0105】

B. シーケンスパラメータセット(`Sequence Parameter Set`、SPS)修正

20

【0106】

表4のように、既存シンタックスのうち一部は、VPSに反映され、SPSから削除されることができる。一方、SPSには`vps_id`シンタックス要素が追加されることができる。`vps_id`が追加されたSPSシンタックスは、表4の通りである。表4において、削除されたシンタックスは、シンタックス中間に線を引いて表現される。

【0107】

`vps_id`は、SPSにより参照されるビデオパラメータセット(`video parameter set`)を識別するための識別子を指示し、`vps_id`は、0~Xの範囲を有することができる。

30

【0108】

【表 4】

	Descriptor
seq_parameter_set_rbsp( ) {	
— profile_ide	u(8)
— reserved_zero_8bits /* equal to 0 */	u(8)
— level_ide	u(8)
seq_parameter_set_id	ue(v)
vps_id	ue(v)
...	
— pie_width_in_luma_samples	ue(v)
— pie_height_in_luma_samples	ue(v)
— pie_cropping_flag	u(1)
— if( pie_cropping_flag ) {	
— pie_crop_left_offset	ue(v)
— pie_crop_right_offset	ue(v)
— pie_crop_top_offset	ue(v)
— pie_crop_bottom_offset	ue(v)
— }	
...	
— max_temporal_layers_minus1	u(3)
— bit_depth_luma_minus8	ue(v)
— bit_depth_chroma_minus8	ue(v)
— chroma_format_ide	u(2)
— temporal_id_nesting_flag	u(1)
...	
— tiles_or_entropy_coding_syne_ide	u(2)
— if( tiles_or_entropy_coding_syne_ide == 1 ) {	
— num_tile_columns_minus1	ue(v)
— num_tile_rows_minus1	ue(v)
— uniform_spacing_flag	u(1)
— if( !uniform_spacing_flag ) {	
— for( i = 0; i < num_tile_columns_minus1; i++ )	
— column_width[ i ]	ue(v)
— for( i = 0; i < num_tile_rows_minus1; i++ )	
— row_height[ i ]	ue(v)
— }	
— loop_filter_across_tiles_enabled_flag	u(1)
— }	
sps_extension_flag	u(1)
if( sps_extension_flag )	
while( more_rbsp_data( ) )	
sps_extension_data_flag	u(1)
rbsp_trailing_bits( )	
}	

10

20

30

40

C. ビデオパラメータセット(Video Parameter Set)のための活性化(activation)シグナリング

【0110】

スライスヘッダには、該当スライスが参照するピクチャパラメータセットに対するインデックス情報が含まれており、ピクチャパラメータセットには、該当ピクチャが参照するシーケンスパラメータセットに対するインデックス情報が含まれている。シーケンスパラメータセットには、該当シーケンスが参照するビデオパラメータセットに対する情報が含まれている。このようにパラメータセットに対する情報をパーシングし、パーシングされた該当パラメータセット情報を参照することを活性化(activation)という。

【0111】

特定パラメータセットに対する情報を利用するために、即ち、パラメータセットを活性化するためにはスライスヘッダから順次にパーシングされなければならない。いずれのSPSが活性化されるか(active)を知るために、全てのスライスヘッダ(slice header)及び関連したPPSが分析されなければならないことを意味する。

【0112】

単一階層を含むビットストリームのうち、サブレイヤ(時間的レイヤ)の中から一部を抽出する時、抽出器(Extractor)は、NALUヘッダと複数のパラメータセット(parameter set)を分析(パーシング)する必要がある。

【0113】

もし、ビデオパラメータセットまたはシーケンスパラメータセットにNALUの抽出のための情報が含まれている場合、抽出器は、スライスヘッダから順次に上位のパラメータセットをパーシングしなければならない。これは抽出器がパラメータセット(parameter set)とスライスヘッダ(slice header)の全てのシンタックス要素を理解しなければならないということの意味する。

【0114】

また、映像のデコーディング過程でも複雑なパーシング過程無しでvp\_s\_idまたはsp\_s\_idを探し、必要なパラメータセットのみを活性化することができる。この場合、ビデオパラメータセットまたはシーケンスパラメータセットが活性化されるパラメータインデックス情報を含む場合、複雑なスライスヘッダ(slice header)及び関連したPPSに対するパーシング手順を減少させることができる。

【0115】

一方、このようなシンタックスの要素のうち、一部分のみがビットストリーム抽出のために必要な情報を含むことができる。それにも拘わらず、抽出器が全てのシンタックス要素を分析することは、大きい負担になることができる。このような問題を解決するために、下記のような方法を提案する。

【0116】

本発明において、パラメータセットの活性化(activation)は、抽出器がスライスヘッダ(slice header)及びそれと関連したPPS(picture parameter set)の分析無しでいずれのパラメータセットが活性化されるかを知ることができるようにシグナリングすることを意味する。

【0117】

本発明によると、いずれのビデオパラメータセット、シーケンスパラメータセットまたはピクチャパラメータセットが活性化されるかどうかを別途にシグナリングすることで、抽出器は、全てのスライスヘッダ(slice header)及び関連したPPSを分析しなければならない負担を減少させることができる。

【0118】

ビデオパラメータセットは、アップデートされることもできる。抽出器がスライスヘッダを分析せずに、現在活性化されるVPS及び関連したSPSまたはPPSを知ることができるように、下記の方法のうち一つが使われることができる。

【0119】

10

20

30

40

50

(1) vps\_id、sps\_id、pps\_idをアクセスユニットデリミタ(access\_unit\_delimiter)に含ませることができる。vps\_id、sps\_id、pps\_idは、各々、関連したAU内のNALUのために使われたビデオパラメータセット、シーケンスパラメータセット、ピクチャパラメータセットの識別子を示す。

【0120】

アクセスユニットデリミタ(Access\_unit\_delimiter)内に各識別子の存在可否を示すために、vps\_id\_present\_flag、sps\_id\_present\_flag、pps\_id\_present\_flagを使用し、提案するアクセスユニットデリミタのシンタックスは、表5の通りである。

【0121】

10

【表5】

access_unit_delimiter_rbsp( ) {	Descriptor
pic_type	u(3)
vps_id_present_flag	u(1)
sps_id_present_flag	u(1)
pps_id_present_flag	u(1)
if( vps_id_present_flag )	
vps_id	ue(v)
if( sps_id_present_flag )	
sps_id	ue(v)
if( pps_id_present_flag )	
pps_id	ue(v)
rbbsp_trailing_bits( )	
}	

20

【0122】

(1-1)他の方法は、表6のように、sps\_id及びpps\_idを除いてvps\_idのみをアクセスユニットデリミタ(access\_unit\_delimiter)に含ませることができる。

30

【0123】

【表6】

access_unit_delimiter_rbsp( ) {	Descriptor
pic_type	u(3)
vps_id	ue(v)
rbbsp_trailing_bits( )	
}	

40

【0124】

(2)ビデオパラメータセットの活性化シグナリングのための他の方法は、新たなSEIメッセージ(parameter\_set\_reference)を使用することである。SEIメッセージは、関連したAU内のNALUのために使われたビデオパラメータセット、シーケンスパラメータセット、ピクチャパラメータセットの識別子を示すvps\_id、sps\_id、pps\_idの存在可否を知らせるためのシンタックスを含む。

【0125】

各識別子の存在可否を示すために、vps\_id\_present\_flag、sps\_id\_present\_flag、pps\_id\_present\_flagシンタックスが使

50

われることができ、SEIシンタックスは、表7の通りである。

【0126】

【表7】

parameter_set_reference( payloadSize ) {	Descriptor
<b>vps_id_present_flag</b>	u(1)
<b>sps_id_present_flag</b>	u(1)
<b>pps_id_present_flag</b>	u(1)
if( vps_id_present_flag )	
<b>vps_id</b>	ue(v)
if( sps_id_present_flag )	
<b>sps_id</b>	ue(v)
if( pps_id_present_flag )	
<b>pps_id</b>	ue(v)
}	

10

【0127】

(2-1)また、表8のようにpps\_idを除いてsps\_idとvps\_idをSEIメッセージに含ませて活性化を知らせることもできる。SEIメッセージに含まれているsps\_idとvps\_idは、該当SEIメッセージと関連したアクセスユニット(access\_unit)のビデオコーディングレイヤNALUが参照するsps\_idとvps\_idを含むことができる。したがって、sps\_idとvps\_idは、活性化される可能性があるパラメータセットの情報を示すことができる。

20

【0128】

【表8】

parameter_set_reference( payloadSize ) {	Descriptor
<b>vps_id</b>	u(4)
<b>sps_id_present_flag</b>	u(1)
if(sps_id_present_flag)	
<b>sps_id</b>	ue(v)
<b>psr_extension_flag</b>	u(1)
if(psr_extension_flag){	
<b>psr_extension_length</b>	ue(v)
for(i=0; i<psr_extension_length; i++)	
<b>psr_extension_data_byte</b>	u(8)
}	
}	

30

40

【0129】

表8において、vps\_idは、現在活性化されるビデオパラメータセットのvideo\_parameter\_set\_idを示す。vps\_id値は、0~15の値を有することができる。

【0130】

sps\_id\_present\_flagが1の値を有する場合、現在活性化されるシーケンスパラメータセットのsequence\_parameter\_set\_idが該当SEIメッセージに含まれていることを示し、sps\_id\_present\_flagが0

50

の値を有する場合、活性化されるシーケンスパラメータセットの `sequence_parameter_set_id` が該当 SEI メッセージに含まれていないことを示す。

【0131】

`sps_id` は、現在活性化されるシーケンスパラメータセットの `sequence_parameter_set_id` を示す。`sps_id` は、0 ~ 31 の値、より限定的には、0 ~ 15 の値を有することができる。

【0132】

`psr_extension_flag` が 0 の場合、`parameter_set_reference SEI message extension` シンタックス要素が `parameter_set_reference SEI message extension` シンタックス要素を `parameter_set_reference SEI message extension` シンタックス要素を拡張して使用することを意味する。

10

【0133】

`psr_extension_length` は、`psr_extension_data` の長さを示す。`psr_extension_length` は、0 ~ 256 の範囲を値を有することができる、`psr_extension_data_byte` は、どのような値も有することができる。

【0134】

(2 - 2) また、表 9 のように、`pps_id` を除いて一つ以上の `sps_id` と `vps_id` を SEI メッセージに含ませてシグナリングすることもできる。

20

【0135】

【表 9】

<code>parameter_set_reference( payloadSize ) {</code>	<b>Descriptor</b>
<b><code>vps_id</code></b>	<code>u(4)</code>
<b><code>num_referenced_sps</code></b>	<code>u(5)</code>
<code>for(i=0; i&lt; num_referenced_sps; i++)</code>	
<b><code>sps_id(i)</code></b>	<code>ue(v)</code>
<code>}</code>	

30

40

【0136】

表 9 において、`vps_id` は、現在活性化されるビデオパラメータセットの `video_parameter_set_id` を示す。`vps_id` は、0 ~ 15 の値を有することができる。

【0137】

`num_reference_sps` は、現在活性化される `vps_id` を参照するシーケンスパラメータセットの個数を示す。

【0138】

`sps_id(i)` は、現在活性化されるシーケンスパラメータセットの `sequence`

50

e\_parameter\_set\_idを示し、sps\_idは、0～31の値、より限定的には、0～15の値を有することができる。

【0139】

(2-3)また、表10のように、sps\_id及びpps\_idを除いてvps\_idのみをSEIメッセージに含ませてシグナリングすることもできる。

【0140】

【表10】

parameter_set_reference( payloadSize ) {	Descriptor
<b>vps_id</b>	ue(v)
}	

10

【0141】

(3)ビデオパラメータセットの活性化シグナリングのための他の方法は、Buffering period SEIメッセージにvps\_id、sps\_id、pps\_idを知らせる情報を含ませることである。表11は、vps\_id、sps\_id、pps\_id識別子の存在可否を示すためのvps\_id\_present\_flag、sps\_id\_present\_flag、pps\_id\_present\_flagを含んでいるシンタックスを示す。

【0142】

【表11】

20

buffering_period( payloadSize ) {	Descriptor
<b>vps_id_present_flag</b>	u(1)
<b>sps_id_present_flag</b>	u(1)
<b>pps_id_present_flag</b>	u(1)
if( vps_id_present_flag )	
<b>vps_id</b>	ue(v)
if( sps_id_present_flag )	
<b>sps_id</b>	ue(v)
if( pps_id_present_flag )	
<b>pps_id</b>	ue(v)
.....	
}	

30

【0143】

(3-1)また、表12のように、sps\_id及びpps\_idを除いてvps\_idのみをBuffering period SEIメッセージに含ませてパラメータセットの活性化をシグナリングすることもできる。

【0144】

40

【表 1 2】

	Descriptor
buffering_period( payloadSize ) {	
<b>vps_id_present_flag</b>	<b>u(1)</b>
if( vps_id_present_flag )	
<b>vps_id</b>	<b>ue(v)</b>
.....	
}	

10

## 【 0 1 4 5】

(4)パラメータセットの活性化シグナリングのための他の方法は、Recovery point SEIメッセージにvps\_id、sps\_id、pps\_idを知らせる情報を含ませることである。表13は、vps\_id、sps\_id、pps\_id識別子の存在可否を示すためのvps\_id\_present\_flag、sps\_id\_present\_flag、pps\_id\_present\_flagを含んでいるシンタックスを示す。

## 【 0 1 4 6】

【表 1 3】

	Descriptor
recovery_point( payloadSize ) {	
vps_id_present_flag	<b>u(1)</b>
sps_id_present_flag	<b>u(1)</b>
pps_id_present_flag	<b>u(1)</b>
if( vps_id_present_flag )	
vps_id	<b>ue(v)</b>
if( sps_id_present_flag )	
sps_id	<b>ue(v)</b>
if( pps_id_present_flag )	
pps_id	<b>ue(v)</b>
recovery_poc_cnt	<b>ue(v)</b>
exact_match_flag	<b>u(1)</b>
broken_link_flag	<b>u(1)</b>
}	
recovery_point( payloadSize ) {	<b>Descriptor</b>
<b>vps_id_present_flag</b>	<b>u(1)</b>
if( vps_id_present_flag )	
<b>vps_id</b>	<b>ue(v)</b>
<b>recovery_poc_cnt</b>	<b>ue(v)</b>
<b>exact_match_flag</b>	<b>u(1)</b>
<b>broken_link_flag</b>	<b>u(1)</b>
}	

20

30

40

## 【 0 1 4 7】

(4-1)また、表14のように、sps\_id及びpps\_idを除いてvps\_idのみをRecovery point SEIメッセージに含ませて知らせる方法もある。

## 【 0 1 4 8】

【表 1 4】

recovery_point( payloadSize ) {	Descriptor
<b>vps_id_present_flag</b>	<b>u(1)</b>
if( vps_id_present_flag )	
<b>vps_id</b>	<b>ue(v)</b>
<b>recovery_poc_cnt</b>	<b>ue(v)</b>
<b>exact_match_flag</b>	<b>u(1)</b>
<b>broken_link_flag</b>	<b>u(1)</b>
}	

10

## 【0149】

前述した `vps_id` または `sps_id` を伝達するメッセージは、`IRAP( intra_random_access_point )` アクセスユニットに含まれることができる。

## 【0150】

前述した情報シグナリング方法のうち少なくとも一つがアクセスユニットに含まれて使われる場合、抽出器は、ビットストリームを抽出するために、前記シグナリング方法を介して `vps_id`、`sps_id`、`pps_id` 値を探し、一つ以上の `vps/sps/pps` を管理することができる。

20

## 【0151】

また、デコーディング装置またはデコーディングを実行するデコーディング部は、前記シグナリング方法を介して `vps_id`、`sps_id`、`pps_id` 値を探し、該当パラメータセットを活性化してパラメータセットと関連したAUをデコーディングすることができる。

## 【0152】

拡張タイプ(Extended type)におけるビットストリームの表現

## 【0153】

以下、階層拡張をサポートするビットストリームを含む場合、スケーラブル階層に対する情報を表示し、これをシグナリングするためのVPSの `extension_info()` と新たなSEIメッセージを提案する。拡張タイプ(Extended type)において、ビットストリームを表現するためには下記のような情報がシグナリングされることができる。

30

## 【0154】

`layer_id` は、レイヤの優先順位(`priority`)値を伝達するかどうかを示すものをシグナリングする。

## 【0155】

このとき、各 `layer_id` 値に対応して空間階層(`dependency_id` 値により識別)、画質階層(`quality_id` 値により識別)、視点(`view_id` 値により識別)などをシグナリングすることができ、時間的レイヤは、NALUヘッダの `temporal_id` により識別されることができる。

40

## 【0156】

また、`layer_id` と関連したビデオの領域は `region_id` によりシグナリングされることができる。

## 【0157】

また、スケーラブル階層のうち、`dependency` 情報、各スケーラブル階層の `bitrate` 情報、各スケーラブル階層の品質情報がシグナリングされることができる。

## 【0158】

`extension_info()` シンタックスは、表15の通りである。

## 【0159】

50

【表 1 5】

extension_info( ) {		
<b>num_frame_sizes_minus1</b>	ue(v)	
<b>num_rep_formats_minus1</b>	ue(v)	
for( i = 0; i <= num_frame_sizes_minus1; i++ ) {		
<b>pic_width_in_luma_samples[ i ]</b>	ue(v)	
<b>pic_height_in_luma_samples[ i ]</b>	ue(v)	
<b>pic_cropping_flag[ i ]</b>	u(1)	
if( pic_cropping_flag[ i ] ) {		10
<b>pic_crop_left_offset[ i ]</b>	ue(v)	
<b>pic_crop_right_offsetv[ i ]</b>	ue(v)	
<b>pic_crop_top_offset[ i ]</b>	ue(v)	
<b>pic_crop_bottom_offset[ i ]</b>	ue(v)	
}		
}		
for( i = 0; i <= num_rep_formats_minus1; i++ ) {		
<b>bit_depth_luma_minus8[ i ]</b>	ue(v)	20
<b>bit_depth_chroma_minus8[ i ]</b>	ue(v)	
<b>chroma_format_idc[ i ]</b>	u(2)	
}		
<b>num_layers_minus1</b>	ue(v)	
<b>dependency_id_flag</b>	u(1)	
<b>quality_id_flag</b>	u(1)	
<b>view_id_flag</b>	u(1)	
<b>region_id_flag</b>	u(1)	
<b>layer_dependency_info_flag</b>	u(1)	30
for( i = 1; i <= num_layers_minus1; i++ ) {		
<b>frame_size_idx[ i ]</b>	ue(v)	
<b>rep_format_idx[ i ]</b>	ue(v)	
if( dependency_id_flag ) {		
<b>one_dependency_id_flag[ i ]</b>	u(1)	
if ( one_dependency_id_flag[ i ] == 1 )		
<b>dependency_id[ i ]</b>	ue(v)	
else if( one_dependency_id_flag[ i ] == 0 ) {		
<b>dependency_id_min[ i ]</b>	ue(v)	40
<b>dependency_id_max[ i ]</b>	ue(v)	
}		
}		
}		

if( quality_id_flag ) {	
<b>one_quality_id_flag</b> [i]	u(1)
if ( one_quality_id_flag[i] == 1)	
<b>quality_id</b> [i]	ue(v)
else if( one_quality_id_flag[i] == 0) {	
<b>quality_id_min</b> [i]	ue(v)
<b>quality_id_max</b> [i]	ue(v)
}	
}	
if( view_id_flag ) {	
<b>one_view_id_flag</b> [i]	u(1)
<b>depth_flag</b> [i]	u(1)
if ( one_view_id_flag[i] == 1)	
<b>view_id</b> [i]	ue(v)
else if( one_view_id_flag[i] == 0) {	
<b>view_id_min</b> [i]	ue(v)
<b>view_id_max</b> [i]	ue(v)
}	
}	
if( region_id_flag ) {	
<b>num_regions_minus1</b> [i]	u(8)
for( j = 1; j <= num_regions_minus1[i]; j++ )	
<b>region_id</b> [i][j]	u(8)
}	
if( layer_dependency_info_flag ) {	
<b>num_directly_dependent_layers</b> [ i ]	u(8)
for( j = 0; j < num_directly_dependent_layers[ i ]; j++ )	
<b>directly_dependent_layer_id_delta_minus1</b> [ i ][ j ]	u(8)
}	
}	
}	

10

20

30

## 【 0 1 6 0 】

表 15 のシンタックスに対する意味は、下記の通りである。

## 【 0 1 6 1 】

- num\_frame\_sizes\_minus1 plus 1 は、符号化されたビデオシーケンス内に含まれている他の種類の映像の大きさ情報(例えば、pic\_width\_in\_luma\_samples[i]、pic\_height\_in\_luma\_samples[i]、pic\_cropping\_flag[i]、pic\_cropping\_flag[i]、pic\_crop\_left\_offset[i]、pic\_crop\_right\_offset[i]、pic\_crop\_top\_offset[i]、pic\_crop\_bottom\_offset[i])の最大個数を示す。num\_frame\_sizes\_minus1 値は、0 ~ X の範囲を有することができる。他の種類の映像とは、異なる解像度を有する映像を含むことができる。

40

## 【 0 1 6 2 】

50

- `num_rep_formats_minus1_plus1` は、符号化されたビデオシーケンス内に含まれている他の種類のビット深さ(`bit_depth`)と色差フォーマット(`chroma_format`)(例えば、`bit_depth_luma_minus8[i]`、`bit_depth_chroma_minus8[i]`、及び`chroma_format_idc_values[i]`)の最大個数を示す。`num_rep_formats_minus1`値は、0～Xの範囲を有する。

【0163】

- `pic_width_in_luma_samples[i]`、`pic_height_in_luma_samples[i]`、`pic_cropping_flag[i]`、`pic_cropping_flag[i]`、`pic_crop_left_offset[i]`、`pic_crop_right_offset_v[i]`、`pic_crop_top_offset[i]`、`pic_crop_bottom_offset[i]`は、符号化されたビデオシーケンスの*i*番目の`pic_width_in_luma_samples`、`pic_height_in_luma_samples`、`pic_cropping_flag`、`pic_cropping_flag`、`pic_crop_left_offset`、`pic_crop_right_offset_v`、`pic_crop_top_offset`、`pic_crop_bottom_offset`値を示す。

10

【0164】

- `bit_depth_luma_minus8[i]`、`bit_depth_chroma_minus8[i]`、及び`chroma_format_idc[i]`は、符号化されたビデオシーケンスの*i*番目の`bit_depth_luma_minus8`、`bit_depth_chroma_minus8`、及び`chroma_format_idc`値を示す。

20

【0165】

- `num_layers_minus1` は、ビットストリームで可能なスケーラブル階層の数を示す。

【0166】

- `dependency_id_flag`が1の場合、`layer_id`値と関連した一つ以上の`dependency_id`値があることを示す。

【0167】

- `quality_id_flag`が1の場合、`layer_id`値と関連した一つ以上の`quality_id`値があることを示す。

30

【0168】

- `view_id_flag`が1の場合、`layer_id`値と関連した一つ以上の`view_id`値があることを示す。

【0169】

- `region_id_flag`が1の場合、`layer_id`値と関連した一つ以上の`region_id`値があることを示す。

【0170】

- `layer_dependency_info_flag`が1の場合、スケーラブル階層の`dependency`情報を提供することを示す。

40

【0171】

- `frame_size_idx[i]`は、`layer_id`値が*i*である階層に適用される`frame_size`のセットに対するインデックスを示す。`frame_size_idx[i]`は、0～X範囲の値を有する。

【0172】

- `rep_format_idx[i]`は、`layer_id`値が*i*である階層に適用されるビット深さ(`bit_depth`)と色差フォーマット(`chroma_format`)のセットに対するインデックスを示す。`rep_format_idx[i]`は、0～X範囲の値を有する。

【0173】

50

- `one_dependency_id_flag[i]`が1の場合、`layer_id`値が*i*と関連した一つの`dependency_id`のみが存在することを示し、`one_dependency_id_flag[i]`が0の場合、`layer_id`値が*i*と関連した二つまたはそれ以上の`dependency_id`値が存在することを示す。

【0174】

- `dependency_id[i]`は、`layer_id`値が*i*と関連した`dependency_id`値を示す。

【0175】

- `dependency_id_min[i]`及び`dependency_id_max[i]`は、`layer_id`値が*i*と関連した最小`dependency_id`値と最大`dependency_id`値を各々示す。

10

【0176】

- `one_quality_id_flag[i]`が1の場合、`layer_id`値が*i*と関連した一つの`quality_id`のみ存在することを示し、`one_quality_id_flag[i]`が0の場合、`layer_id`値が*i*と関連した二つまたはそれ以上の`quality_id`値が存在することを示す。

【0177】

- `quality_id[i]`は、`layer_id`値が*i*と関連した`quality_id`値を示す。

【0178】

20

- `quality_id_min[i]`及び`quality_id_max[i]`は、`layer_id`値が*i*と関連した最小`quality_id`値と最大`quality_id`値を各々示す。

【0179】

- `one_view_id_flag[i]`が1の場合、`layer_id`値が*i*と関連した一つの`view_id`が存在することを示し、0の場合、`layer_id`値が*i*と関連した二つまたはそれ以上の`view_id`値が存在することを示す。

【0180】

- `view_id[i]`は`layer_id`値が*i*と関連した`view_id`値を示す。

【0181】

30

- `depth_flag[i]`が1の場合、`layer_id`値が*i*である現在スケーラブル階層が3Dビデオビットストリームの深さ情報を含んでいることを示す。

【0182】

- `view_id_min[i]`及び`view_id_max[i]`は、`layer_id`値が*i*と関連した最小`view_id`値と最大`view_id`値を各々示す。

【0183】

- `num_regions_minus_1_plus_1`は、`layer_id`値が*i*と関連した領域の個数を示す。

【0184】

- `region_id[j]`は、`layer_id`値が*i*と関連した領域*j*の識別子を示す。

40

【0185】

- `num_directly_dependent_layers[i]`は、現在スケーラブル階層*i*が直接的に関連したスケーラブル階層(デコーディング時、予測信号の形成に必要な階層)の数を示す。

【0186】

- `directly_dependent_layer_id_delta_minus_1[i][j] plus_1`は、現在スケーラブル階層である`layer_id[i]`と、現在スケーラブル階層が直接的に関連した*j*番目のスケーラブル階層の階層識別子との間の差を示す。*j*番目の直接的に関連したスケーラブル階層の階層識別子は、(`layer_id[i]`

50

- `directly_dependent_layer_id_delta_minus_1[i][j]-1`)である。

【0187】

他の実施例に係る `extension_info()` シンタックスは、表16の通りである。

【0188】

【表 1 6】

extension_info( ) {		
<b>num_frame_sizes_minus1</b>	ue(v)	
<b>num_rep_formats_minus1</b>	ue(v)	
for( i = 0; i <= num_rep_formats_minus1; i++ ) {		
<b>pic_width_in_luma_samples[ i ]</b>	ue(v)	
<b>pic_height_in_luma_samples[ i ]</b>	ue(v)	10
<b>bit_depth_luma_minus8[ i ]</b>	ue(v)	
<b>bit_depth_chroma_minus8[ i ]</b>	ue(v)	
<b>chroma_format_idc[ i ]</b>	u(2)	
<b>pic_cropping_flag[ i ]</b>	u(1)	
if( pic_cropping_flag[ i ] ) {		
<b>pic_crop_left_offset[ i ]</b>	ue(v)	
<b>pic_crop_right_offset[ i ]</b>	ue(v)	
<b>pic_crop_top_offset[ i ]</b>	ue(v)	
<b>pic_crop_bottom_offset[ i ]</b>	ue(v)	20
}		
}		
<b>num_layers_minus1</b>	ue(v)	
<b>dependency_id_flag</b>	u(1)	
<b>quality_id_flag</b>	u(1)	
<b>view_id_flag</b>	u(1)	
<b>region_id_flag</b>	u(1)	
<b>layer_dependency_info_flag</b>	u(1)	30
for( i = 1; i <= num_layers_minus1; i++ ) {		
<b>frame_size_idx[ i ]</b>	ue(v)	
<b>rep_format_idx[ i ]</b>	ue(v)	
if( dependency_id_flag ) {		
<b>one_dependency_id_flag[ i ]</b>	u(1)	
if ( one_dependency_id_flag[ i ] == 1 )		
<b>dependency_id[ i ]</b>	ue(v)	
else if( one_dependency_id_flag[ i ] == 0 ) {		40
<b>dependency_id_min[ i ]</b>	ue(v)	
<b>dependency_id_max[ i ]</b>	ue(v)	
}		

## 【 0 1 8 9 】

表 1 6 に示すように、pic\_width\_in\_luma\_samples[ i ] 及び pic\_height\_in\_luma\_samples[ i ]、bit\_depth\_luma\_minus8[ i ]、bit\_depth\_chroma\_minus8[ i ]、及び chroma\_format\_idc[ i ] は、異なるリプレゼンテーションフォーマットに対する情報

にシグナリングされることができる。

【0190】

他の実施例によると、`pic_width_in_luma_samples[i]`、`pic_height_in_luma_samples[i]`、`bit_depth_luma_minus8[i]`、`bit_depth_chroma_minus8[i]`、及び`chroma_format_idc[i]`は、異なる映像、即ち、異なる解像度を有するピクチャに対する情報にシグナリングされることができる。

【0191】

ビットレートとクオリティ情報のシグナリングのための活性化SEIメッセージに対するシンタックスは、表17の通りである。

【0192】

【表17】

bitrate_quality_info( payloadSize ) {	Descriptor
<b>num_layers_minus1</b>	ue(v)
<b>bitrate_info_flag</b>	u(1)
<b>quality_info_flag</b>	u(1)
<b>quality_type_flag</b>	u(1)
for( i = 1; i <= num_layers_minus1; i++ ) {	
if( bitrate_info_flag ) {	
<b>max_bitrate[i]</b>	u(8)
<b>average_bitrate[i]</b>	u(8)
}	
if( quality_info_flag ) {	
<b>quality_value[i]</b>	u(8)
}	
}	
if( quality_type_flag ) {	
<b>QualityTypeUriIdx = 0</b>	
do	
<b>quality_type_uri[ QualityTypeUriIdx ]</b>	b(8)
<b>while( quality_type_uri[ QualityTypeUriIdx++ ] != 0 )</b>	
}	
}	

【0193】

表17のシンタックスに対する意味は、下記の通りである。

【0194】

- `num_layers_minus1`は、ビットストリームで提供可能なスケーラブル階層の数を示す。

【0195】

- `bitrate_info_flag`が1の場合、各々のスケーラブル階層に対するビットレート情報が提供されることを指示する。

【0196】

- `quality_info_flag`が1の場合、各々のスケーラブル階層に対するクオリティ値に対する情報が提供されることを示す。

【0197】

10

20

30

40

50

- `quality_type_flag` が 1 の場合、各々のスケーラブル階層に対するクオリティタイプに対する情報が提供されることを示す。

【0198】

- `max_bitrate[i]` は、`layer_id` 値が `i` であるスケーラブル階層の最大ビットレートを示し、`average_bitrate[i]` は、`layer_id` 値が `i` であるスケーラブル階層の平均ビットレートを示す。

【0199】

- `quality_value[i]` は、スケーラブル階層 `i` のクオリティ値を示す。

【0200】

- `quality_type_uri[QualityTypeUriIdx]` は、UTF-8 character で符号化された `null_terminated_string` の `QualityTypeUriIdx - th` バイト (byte) であり、クオリティ値のタイプに対する表現を含む URI (universal resource identifier) を示す。

【0201】

以下、効率的なビットストリーム抽出のための記述方式の改善として、VPS (video parameter set) を改善させる方を提案する。

【0202】

階層参照 (Layer Referencing)

【0203】

複数の階層をサポートするビットストリームで `layer_id` とスケーラビリティ次元 ID (scalability dimension ID) との関係を示す方法として、`layer_id` とスケーラビリティ次元 ID (scalability dimension ID) との間のマッピング方法を知らせる第 1 の方法と、`layer_id` のビットを分割 (partitioning または splicing) して割り当てられたビットにどのような次元タイプが存在するかを知らせる第 2 の方法とが存在できる。

【0204】

複数の階層をサポートするビットストリームで次元タイプ (dimension type) とは、空間的スケーラビリティ、クオリティ的スケーラビリティのようなスケーラビリティのタイプを意味し、次元 ID (dimension ID) は、特定の次元タイプが有することができるレイヤに対するインデックスを意味する。

【0205】

複数の階層をサポートするビットストリームで、特定次元 (dimension) では特定階層 (理解を助けるために、例えば、単一階層のビットストリームで時間的スケーラビリティをサポートする場合、時間的レイヤ (sub-layer) 3) が次の低い階層 (例えば、時間的レイヤ (sub-layer)) を直接的に参照することは一般的である。

【0206】

また、例えば、空間スケーラビリティをサポートする場合は、空間レイヤ 2 が次の低い空間階層 1 を直接的に参照することを意味する。

【0207】

したがって、前記のような場合を示すために、基本参照 (default direct dependency) を有する次元を先に記述することを提案する。

【0208】

その後、特定連関性 (dependency) を階層 (scalable layer) に対する説明パート (description loop) で具体的に記述することができる。

【0209】

以下、前記二つの方法を利用して階層参照に対するシグナリングをするための方を提示する。vps\_extension のための改善されたシンタックスは、表 18 乃至表 21 の通りである。

10

20

30

40

50

【 0 2 1 0 】

【 表 1 8 】

	Descriptor	
vps_extension( ) {		
while( !byte_aligned( ) )		
<b>vps_extension_byte_alignment_reserved_one_bit</b>	u(1)	
<b>all_default_dependency_flag</b>	u(1)	
if( all_default_dependency_flag == 0 ) {		
<b>num_default_dim_minus1</b>	u(4)	10
for( j = 0; j <= num_default_dim_minus1; j++ ) {		
<b>dimension[ j ]</b>	u(4)	
}		
}		
for( i = 1; i <= vps_max_layers_minus1; i++ ) {		
// mapping of layer ID to scalability dimension IDs		
<b>num_dimensions_minus1[ i ]</b>	u(4)	
for( j = 0; j <= num_dimensions_minus1; j++ ) {		
<b>dimension_type[ i ][ j ]</b>	u(4)	20
<b>dimension_id[ i ][ j ]</b>	u(8)	
}		
// layer dependency		
<b>specific_dependency_flag[ i ]</b>	u(1)	
if( specific_dependency_flag[ i ] == 1 ) {		
<b>num_direct_ref_layers[ i ]</b>	u(6)	
for( j = 0; j < num_direct_ref_layers[ i ]; j++ )		
<b>ref_layer_id[ i ][ j ]</b>	u(6)	
}		30
}		
}		
}		

【 0 2 1 1 】

表 1 8 は、第 1 の方法を利用して `layer_id` とスケーラビリティ次元 ID ( `scalability dimension ID` ) をマッピングさせているシンタックスを示している。表 1 8 のシンタックスに対する意味は、下記の通りである。

【 0 2 1 2 】

- `all_default_dependency_flag` が 1 の場合、全ての階層次元が基本参照 ( `default dependency` ) を有することを指示する。即ち、特定次元 `i` において、`dimension_id[ i ] = n` である階層は、デフォルトとして `dimension_id[ i ] = n - 1` を有する他の階層を直接的に参照することを意味する。 40

【 0 2 1 3 】

`all_default_dependency_flag` が 0 の場合、全ての階層次元が基本参照を有するものではないことを示す。`all_default_dependency_flag` が 0 の場合、下の `num_default_dim_minus1` がシグナリングされる。

【 0 2 1 4 】

- `num_default_dim_minus1` は、基本参照 ( `default dep` ) 50

`endency`)を有する次元の数を示す。

【0215】

- `dimension[j]`は、基本参照(`default dependency`)を有する階層次元のタイプを明示する。即ち、基本参照(`default dependency`)を有する次元の数を一つずつ増加させながら、基本参照を有する階層次元のタイプに対する情報がシグナリングされる。該当次元において、上位階層(例えば、`dimension_id = n`)は、次の下位階層(例えば、`dimension_id = n - 1`)を直接的に参照する。

【0216】

- `specific_dependency_flag[i]`が1の場合、該当階層のために具体的に記述された直接参照(`direct dependences/references`)があることを意味する。したがって、`specific_dependency_flag[i]`が1の場合、該当階層が直接参照するレイヤの個数と該当レイヤのIDがシグナリングされる。

10

【0217】

階層Cが階層Bを直接的に参照するということは、階層Cを復号化するために、デコーダは、階層Bの情報(デコーディングされ、またはデコーディングされない)を使用しなければならないという意味である。しかし、もし、階層Bが直接的に階層Aの情報を使用する場合、階層Cは階層Aを直接的に参照すると判断されない。

【0218】

20

【表 19】

vps_extension( ) {	Descriptor	
while( !byte_aligned( ) )		
<b>vps_extension_byte_alignment_reserved_one_bit</b>	u(1)	
<b>all_default_dependency_flag</b>	u(1)	
if( all_default_dependency_flag == 0 ) {		
<b>num_default_dim_minus1</b>	u(4)	
for( j = 0; j <= num_default_dim_minus1; j++ ) {		
<b>dimension[ j ]</b>	u(4)	10
}		
}		
// scalability type and layer_id partitioning method		
<del>scalability_type</del>	<del>u(4)</del>	
for( i = 0; i < MaxDim( scalability_type ); i++ )		
<b>num_dimensions_minus1</b>		
for( i = 0; i <= num_dimensions_minus1; i++ ) {		
<b>dimension_type[ i ]</b>	u(4)	20
<b>layer_id_dim_len[ i ]</b>	u(3)	
}		
for( i = 0; i <= max_num_layers_minus1; i++ ) {		
<b>vps_layer_id[ i ]</b>	u(6)	
// layer dependency		
<b>specific_dependency_flag[i]</b>	u(1)	
if( specific_dependency_flag[i] == 1 ) {		
<b>num_direct_ref_layers[ i ]</b>	u(6)	
for( j = 0; j < num_direct_ref_layers[ i ]; j++ )		
<b>ref_layer_id[ i ][ j ]</b>	u(6)	30
}		
}		
}		

## 【0219】

表19は、第2の方法を利用してlayer\_idのビットをスケーラビリティ次元タイプに割り当て、割り当てられた次元タイプの長さをシグナリングするシンタックスを示している。

## 【0220】

表19に含まれているnum\_dimensions\_minus1は、NALUヘッダ内に存在する階層次元の数を示す。即ち、NALUヘッダに存在する階層次元の数を把握し、該当階層次元毎に存在する階層タイプと次元タイプに割り当てられたビット数を把握する。

## 【0221】

表19のシンタックス階層参照のためのシンタックスall\_default\_dependency\_flag、num\_default\_dim\_minus1、dimension[j]及びspecific\_dependency\_flag[i]に対する説明は、表18に含まれているシンタックスと同じ意味を有する。

## 【0222】

10

20

30

40

50

表 2 0 及び表 2 1 は、表 1 8 及び表 1 9 と異なる方式のシンタックスを示す。表 2 0 は、第 1 の方法を利用する場合、基本参照 ( d e f a u l t d e p e n d e n c y ) を示す他のシンタックスを示し、表 2 1 は、第 2 の方法を利用する場合、基本参照 ( d e f a u l t d e p e n d e n c y ) を示す他のシンタックスを示す。

【 0 2 2 3 】

【表 2 0】

vps_extension( ) {	Descriptor	
while( !byte_aligned( ) )		
<b>vps_extension_byte_alignment_reserved_one_bit</b>	u(1)	10
<b>num_dimensions_minus1</b>	u(4)	
for( i = 0; i <= num_dimensions_minus1; i++ ) {		
<b>dimension_type[ i ]</b>	u(4)	
<b>default_dependency_flag [i]</b>	u(1)	
}		
for( i = 1; i <= vps_max_layers_minus1; i++ ) {		
// mapping of layer ID to scalability dimension IDs		
for( j = 0; j <= num_dimensions_minus1; j++ ) {		
<b>dimension_id[ i ][ j ]</b>	u(8)	20
}		
// layer dependency		
<b>specific_dependency_flag [i]</b>	u(1)	
if( <b>specific_dependency_flag [i]</b> == 1 ) {		
<b>num_direct_ref_layers[ i ]</b>	u(6)	
for( j = 0; j < num_direct_ref_layers[ i ]; j++ )		
<b>ref_layer_id[ i ][ j ]</b>	u(6)	
}		
}		30
}		
}		

【 0 2 2 4 】

【表 2 1】

vps_extension( ) {	Descriptor
while( !byte_aligned( ) )	
<b>vps_extension_byte_alignment_reserved_one_bit</b>	u(1)
// scalability type and layer_id partitioning method	
<b>num_dimensions_minus1</b>	
for( i = 0; i <= num_dimensions_minus1; i++ ) {	
<b>dimension_type[ i ]</b>	u(4)
<b>layer_id_dim_len[ i ]</b>	u(3)
<b>default_dependency_flag [i]</b>	u(1)
}	
for( i = 0; i <= max_num_layers_minus1; i++ ) {	
<b>vps_layer_id[ i ]</b>	u(6)
// layer dependency	
<b>specific_dependency_flag [i]</b>	u(1)
if( <b>specific_dependency_flag [i] == 1</b> ) {	
<b>num_direct_ref_layers[ i ]</b>	u(6)
for( j = 0; j < num_direct_ref_layers[ i ]; j++ )	
<b>ref_layer_id[ i ][ j ]</b>	u(6)
}	
}	
}	

10

20

## 【 0 2 2 5 】

表 2 0 及び表 2 1 のシンタックスのうち、表 1 8 及び表 1 9 と重複するシンタックスに対する説明は省略する。

## 【 0 2 2 6 】

表 2 0 及び表 2 1 に含まれている新たなシンタックス `default_dependency_flag[i]` は、次元タイプ `i` が基本参照を使用するかどうかを示す。該当次元において、高い階層(例えば、`dimension_id[i] = n`)は、下の階層(例えば、`dimension_id[i] = n - 1`)を直接的に参照する。

30

## 【 0 2 2 7 】

即ち、`num_dimensions_minus1` と `dimension_type[i]` によって特定次元タイプが指定される場合、該当次元タイプが基本参照を使用するかどうかをシグナリングし、そうでない場合、該当階層が直接的に参照するレイヤに対する情報をシグナリングする。

## 【 0 2 2 8 】

本発明による次元タイプ(`dimensions_type`)を示す場合、表 2 2 の通りである。

40

## 【 0 2 2 9 】

【表 2 2】

dimension_type[ i ][ j ]	dimension_id[ i ][ j ]
0	view order idx
1	depth flag
2	dependency ID
3	quality ID
4	priority ID
5	region ID
6..15	Reserved

10

## 【0230】

本発明によると、既存の次元タイプで次元タイプ4及び5、即ち、priority ID及びregion IDを示すタイプが追加された。

## 【0231】

dimension\_type[ i ][ j ]は、基本的に0～5の値を有することができる。他の値は、以後に定義されることができ、デコーダは、0～5の値でない場合、dimension\_type[ i ][ j ]の値を無視することができる。

## 【0232】

dimension\_typeが4の値を有する場合、該当dimension\_idは、SVC標準でビットストリームのpriority階層のidを示す。

20

## 【0233】

dimension\_typeが5の値を有する場合、該当dimension\_idは、ビットストリームの特定領域のidを示す。特定領域は、ビットストリーム内で一つ以上の時空間セグメント(spatial-temporal segment)になることができる。

## 【0234】

図4は、本発明による映像情報のエンコーディング方法を説明するための制御流れ図である。

30

## 【0235】

図示されているように、符号化装置は、映像に関連した情報を含むNAL(Network Abstraction Layer)ユニットをエンコーディングする(S401)。

## 【0236】

NALユニットのNALユニットヘッダは、NALユニットが非参照ピクチャの少なくとも一部または全部を含むスライスを含むかどうかを示す情報を含まない。

## 【0237】

一方、NALユニットヘッダは、スケーラブル階層をサポートするビットストリームでスケーラブル階層を識別するための階層識別情報を含んでいる。

40

## 【0238】

このとき、NALユニットヘッダに含まれないNALユニットが非参照ピクチャの少なくとも一部または全部を含むスライスを含むかどうかを示す情報をシグナリングするために使われたビットは、階層識別情報をシグナリングするために使われることができる。

## 【0239】

また、NALユニットは、映像のデコーディングのために必要な多様な前記パラメータセットに対する情報を含むことができる。

## 【0240】

符号化装置は、活性化されるパラメータセットに対する情報を含むSEI(Supplemental enhancement information)メッセージを独立さ

50

れたNALユニットにエンコーディングすることができる。

【0241】

活性化されるパラメータセットに対する情報は、活性化されるビデオパラメータセットをインデクシングする情報及び活性化されるシーケンスパラメータセットをインデクシングする情報のうち少なくとも一つを含むことができる。

【0242】

また、活性化されるパラメータセットに対する情報は、活性化されるビデオパラメータセットをインデクシングする情報、活性化されるビデオパラメータセットを参照するシーケンスパラメータセットの個数を示す情報、及びシーケンスパラメータセットをインデクシングする情報を含むことができる。

10

【0243】

このようなパラメータセットに対する情報は、復号化装置が時間的スケーラビリティを提供するサブレイヤを抽出する時に利用されることができる。

【0244】

また、デコーディング装置またはデコーディングを実行するデコーディング部は、ビデオコーディングレイヤNALUのデコーディング時に必要なパラメータセットを活性化する時、前記パラメータセットに対する情報を利用することができる。

【0245】

符号化装置は、エンコーディングされた映像に関連した情報を含むNALユニットをビットストリームに送信する(S402)。

20

【0246】

図5は、本発明による映像情報のデコーディング方法を説明するための制御流れ図である。

【0247】

図5を参照すると、復号化装置は、ビットストリームを介してエンコーディングされた映像に関連した情報を含むNALユニットを受信する(S501)。

【0248】

復号化装置は、NALユニットのヘッダ及びNALペイロード(payload)をパーシングする(S502)。映像情報に対するパーシングは、エントロピー復号化部または別途のパーシング部で実行されることができる。

30

【0249】

復号化装置は、パーシングを介してNALユニットヘッダ及びNALペイロードに含まれている多様な情報を取得することができる。

【0250】

NALユニットヘッダは、スケーラブル階層をサポートするビットストリームでスケーラブル階層を識別するための階層識別情報を含み、NALユニットのエンコーディング時、全体ビットストリームで非参照ピクチャであるか、または参照ピクチャであるかを示す1ビットのフラグ情報を含まない。

【0251】

このとき、NALユニットヘッダに含まれないNALユニットが非参照ピクチャの少なくとも一部または全部を含むスライスを含むかどうかを示す情報をシグナリングするために使われたビットは、階層識別情報をシグナリングするために使われることができる。

40

【0252】

また、復号化装置は、パーシングを介してSEIメッセージに含まれている該当SEIメッセージと関連したNALUをデコーディングするために必要なパラメータセットに対する情報を取得することができる。

【0253】

活性化されるパラメータセットに対する情報は、活性化されるビデオパラメータセットをインデクシングする情報及び活性化されるシーケンスパラメータセットをインデクシングする情報のうち少なくとも一つを含むことができる。

50

【0254】

また、活性化されるパラメータセットに対する情報は、活性化されるビデオパラメータセットをインデクシングする情報、活性化されるビデオパラメータセットを参照するシーケンスパラメータセットの個数を示す情報、及びシーケンスパラメータセットをインデクシングする情報を含むことができる。

【0255】

このようなパラメータセットに対する情報は、復号化装置が時間的スケーラビリティを提供するサブレイヤを抽出する時に利用されることができる。

【0256】

追加的にパラメータセットに対する情報は、ビットストリームを復号化する時またはセッションネゴシエーション(例えば、IP網におけるストリーミング時、session negotiation)する時に利用されることができる。

【0257】

前述した実施例において、方法は一連のステップまたはブロックで流れ図に基づいて説明されているが、本発明は、ステップの順序に限定されるものではなく、あるステップは、前述と異なるステップと、異なる順序にまたは同時に発生できる。また、当該技術分野において、通常の知識を有する当業者であれば、流れ図に示すステップが排他的でなく、他のステップが含まれ、または流れ図の一つまたはそれ以上のステップが本発明の範囲に影響を及ぼさずに削除可能であることを理解することができる。

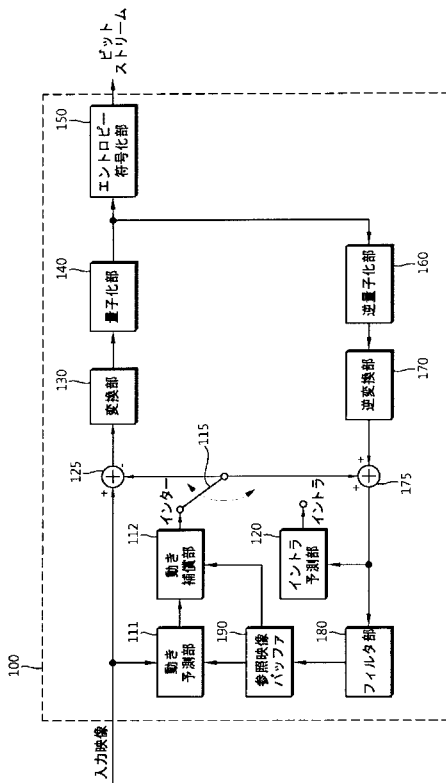
【0258】

前述した実施例は、多様な態様の例示を含む。多様な態様を示す全ての可能な組合せを記述することはできないが、当該技術分野の通常の知識を有する者であれば、他の組合せが可能であることを認識することができる。したがって、本発明は、特許請求の範囲内に属する全ての交替、修正、及び変更を含む。

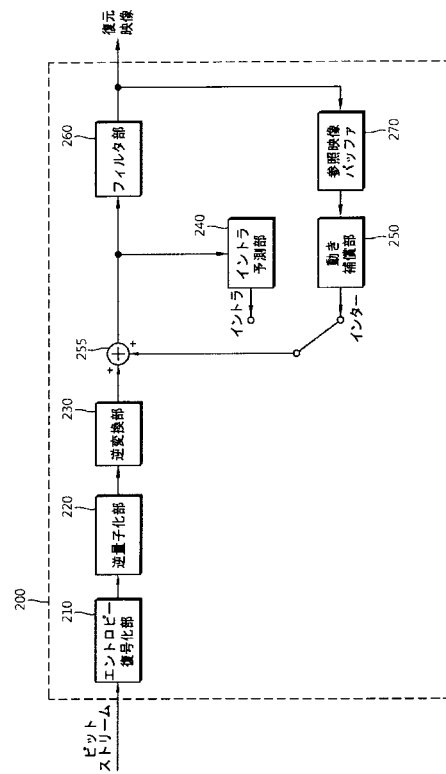
10

20

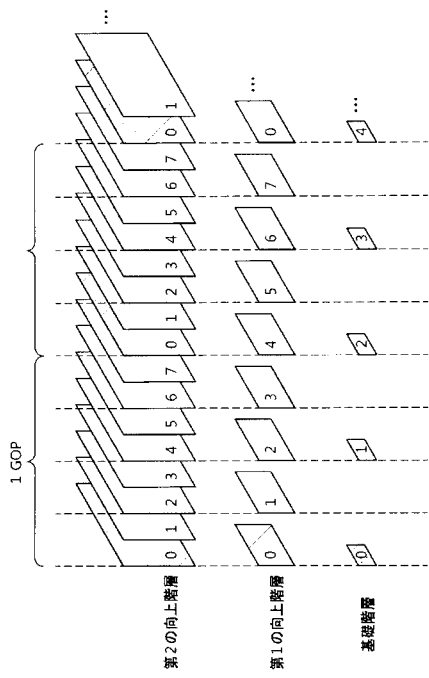
【図1】



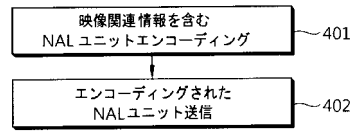
【図2】



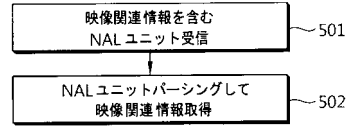
【 図 3 】



【 図 4 】



【 図 5 】



## フロントページの続き

- (31)優先権主張番号 10-2012-0071933  
(32)優先日 平成24年7月2日(2012.7.2)  
(33)優先権主張国 韓国(KR)
- (31)優先権主張番号 10-2012-0077012  
(32)優先日 平成24年7月16日(2012.7.16)  
(33)優先権主張国 韓国(KR)
- (31)優先権主張番号 10-2012-0108925  
(32)優先日 平成24年9月28日(2012.9.28)  
(33)優先権主張国 韓国(KR)
- (31)優先権主張番号 10-2012-0112598  
(32)優先日 平成24年10月10日(2012.10.10)  
(33)優先権主張国 韓国(KR)
- (31)優先権主張番号 10-2013-0041862  
(32)優先日 平成25年4月16日(2013.4.16)  
(33)優先権主張国 韓国(KR)
- (74)代理人 100105153  
弁理士 朝倉 悟
- (74)代理人 100152205  
弁理士 吉田 昌司
- (72)発明者 カン、ジョン、ウォン  
大韓民国テジョン、ユソン - グ、クワンピョン - ドン、ウナム、ネオミア、609 - 501
- (72)発明者 リー、ハ、ヒョン  
大韓民国ソウル特別市、チュンナン - グ、ミョンモク、2 - ドン、136 - 5
- (72)発明者 チェ、ジン、ス  
大韓民国テジョン、ユソン - グ、パンソク - ドン、613、パンソクマウル、6、ダンジ、アパート、609 - 1605
- (72)発明者 キム、ジン、ウン  
大韓民国テジョン、ユソン - グ、ジョンミン - ドン、エキスポ、アパート、305 - 1603
- Fターム(参考) 5C159 MA04 MA05 MA21 MA31 MC11 MC38 ME01 NN01 NN28 PP04  
RB09 RB16 RC11 UA02 UA05 UA33