

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号
特許第6675979号
(P6675979)

(45) 発行日 令和2年4月8日 (2020. 4. 8)

(24) 登録日 令和2年3月13日 (2020. 3. 13)

(51) Int. Cl.

F I

HO 4 N 19/70 (2014. 01) HO 4 N 19/70

HO 4 N 19/30 (2014. 01) HO 4 N 19/30

HO 4 N 19/597 (2014. 01) HO 4 N 19/597

請求項の数 14 (全 36 頁)

(21) 出願番号	特願2016-526589 (P2016-526589)	(73) 特許権者	515089080
(86) (22) 出願日	平成26年7月15日 (2014. 7. 15)		ジーイー ビデオ コンプレッション エ
(65) 公表番号	特表2016-527795 (P2016-527795A)		ルエルシー
(43) 公表日	平成28年9月8日 (2016. 9. 8)		アメリカ合衆国 ニューヨーク州 1 2 2
(86) 国際出願番号	PCT/EP2014/065182		1 1 オールバニー サウスウッズ プー
(87) 国際公開番号	W02015/007751		ルバード 8
(87) 国際公開日	平成27年1月22日 (2015. 1. 22)	(74) 代理人	100079577
審査請求日	平成28年3月8日 (2016. 3. 8)		弁理士 岡田 全啓
審査番号	不服2018-8825 (P2018-8825/J1)	(72) 発明者	ズーリング カルステン
審査請求日	平成30年6月27日 (2018. 6. 27)		ドイツ連邦共和国 1 0 2 4 7 ベルリン
(31) 優先権主張番号	61/846, 479		シュライナーシュトラーセ 6 4 アー
(32) 優先日	平成25年7月15日 (2013. 7. 15)	(72) 発明者	シーアル トーマス
(33) 優先権主張国・地域又は機関	米国 (US)		ドイツ連邦共和国 1 0 4 3 7 ベルリン
			ドゥンカーシュトラーセ 7 2

最終頁に続く

(54) 【発明の名称】 多層符号化における層特徴の信号

(57) 【特許請求の範囲】

【請求項 1】

レシーバー (1 0 2) と層識別エクステンダー (1 1 0) とを含み、
前記レシーバー (1 0 2) は、それぞれがベース層 I D フィールド (1 0 8) と拡張層 I D フィールド (1 1 2) とで構成された層識別構文要素構造を含む一連のパケット (1 0 6) から成る多層ビデオ信号 (1 0 4) を受信するように構成され、
前記層識別エクステンダー (1 1 0) は、複数のパケットを含む前記多層ビデオ信号の予め決められた部分 (1 1 3) について、前記多層ビデオ信号の前記予め決められた部分 (1 1 3) に含まれる前記パケットの拡張層 I D フィールド (1 1 2) の仮の最大値を示す最大構文要素 (1 1 4) を前記多層ビデオ信号から読み取り、
前記多層ビデオ信号の前記予め決められた部分内の各パケットについて、前記層識別構文要素構造に基づいて前記各パケットのための層 I D を決定し、
前記仮の最大値に基づいて、前記多層ビデオ信号の前記予め決められた部分内の層の最大数を決定し、そして、
前記多層ビデオ信号の層特徴付け構文部分 (1 2 4) を層の最大数と等しい回数繰り返し返して構文解析することによって、最大数の層のそれぞれについて最低 1 つの特徴を決定するように構成され、
前記装置は、ビットストリームの中の前記最大構文要素 (1 1 4) から前記拡張層 I D フィールド (1 1 2) のビット長さを得るように構成されることを特徴とする装置。

【請求項 2】

別の層から相互層予測されたいずれかの層が、更なるビュー、奥行き情報、アルファ混合情報、カラーコンポーネント情報、空間の解像度改良およびS/NR解像度改良のうちの1つ以上を追加するように、前記多層ビデオ信号が相互層予測を使用して異なる層でビデオ素材を符号化していること、を特徴とする請求項1に記載の装置。

【請求項3】

前記パケット(106)が、NALユニット、スライス、タイル及び画像で構成されるグループのうちの1つであること、を特徴とする請求項1または請求項2に記載の装置。

【請求項4】

前記仮の最大値を使って、前記拡張層IDフィールドのビット長さを引き出すように構成されていること、を特徴とする請求項1～請求項3のいずれかに記載の装置。

10

【請求項5】

2を底とする前記仮の最大値の対数を切り上げることによって、前記拡張層IDフィールドのビット長さを引き出すように構成されていること、を特徴とする請求項1～請求項4のいずれかに記載の装置。

【請求項6】

前記最低1つの特徴が、前記層の最大数のうちの他いずれかの層に対する前記各層の相互層予測依存と関連すること、を特徴とする請求項1～請求項5のいずれかに記載の装置。

【請求項7】

ビットストリームの中の明示的な信号から前記拡張層IDフィールド(112)のビット長さを得るように構成される装置であって、前記最大構文要素(114)は、前記拡張層IDフィールド(112)の仮の最大値を前記多層ビデオ信号の予め決められた部分(113)ごとに示すこと、を特徴とする請求項1～請求項6のいずれかに記載の装置。

20

【請求項8】

前記各パケットについての前記層IDを、前記ベース層IDフィールドと前記拡張層IDフィールドとを連結することによって、前記層識別構文要素構造に基づいて決定するように構成されていること、を特徴とする請求項1～請求項6のいずれかに記載の装置。

【請求項9】

各層に対して決定された前記最低1つの特徴を使った相互層予測を使って、前記多層ビデオ信号を復号化するように構成されたビデオデコーダ、もしくは、

30

各層に対して決定された前記最低1つの特徴に基づいて、前記多層ビデオ信号のパケットを除去するように構成されたネットワーク要素であること、を特徴とする請求項1～請求項8のいずれかに記載の装置。

【請求項10】

それぞれが、ベース層IDフィールド(108)と拡張層IDフィールド(112)とで構成された層識別構文要素構造を含む一連のパケット(106)から成る多層ビデオ信号(104)を受信し、

複数のパケットを含む前記多層ビデオ信号の予め決められた部分(113)について、前記多層ビデオ信号の前記予め決められた部分(113)に含まれる前記パケットの拡張層IDフィールド(112)の仮の最大値を示す最大構文要素(114)を前記多層ビデオ信号から読み取り、

40

前記多層ビデオ信号の前記予め決められた部分内の各パケットについて、前記層識別構文要素構造に基づいて前記各パケットのための層IDを決定し、

前記仮の最大値に基づいて、前記多層ビデオ信号の前記予め決められた部分内の層の最大数を決定し、そして、

前記多層ビデオ信号の層特徴付け構文部分(124)を層の最大数と等しい回数繰り返して構文解析することによって、最大数の層のそれぞれについて最低1つの特徴を決定することを含む方法であって、

ビットストリームの中の前記最大構文要素(114)から前記拡張層IDフィールド(112)のビット長さを得ることを含むことを特徴とする方法。

50

【請求項 1 1】

それぞれが、ベース層 ID フィールド (1 0 8) と拡張層 ID フィールド (1 1 2) とで構成された層識別構文要素構造を含む一連のパケット (1 0 6) から成る多層ビデオ信号にビデオ信号を符号化するためのエンコードであって、前記エンコードは、

複数のパケットを含む前記多層ビデオ信号の予め決められた部分 (1 1 3) について、前記多層ビデオ信号の前記予め決められた部分 (1 1 3) に含まれる前記パケットの拡張層 ID フィールド (1 1 2) の仮の最大値を示す最大構文要素 (1 1 4) を、前記多層ビデオ信号に挿入し、

前記仮の最大値に基づいて、前記多層ビデオ信号の前記予め決められた部分内の層の最大数を決定し、そして、

前記多層ビデオ信号の層特徴付け構文部分 (1 2 4) を層の最大数と等しい回数繰り返して書き込むことによって、前記多層ビデオ信号 (1 0 4) の最大数の層のそれぞれについて最低 1 つの特徴を信号で伝えるように構成されたエンコードであって、

前記エンコードは、前記最大構文要素 (1 1 4) によって、前記ビットストリームの中の前記拡張層 ID フィールド (1 1 2) のビット長さを信号で伝えることを特徴とするエンコード。

【請求項 1 2】

2 を底とする前記仮の最大値の対数を切り上げることによって、前記拡張層 ID フィールドのビット長さが前記追加の構文要素から引き出されるようにして、前記追加の構文要素を引き出すように構成されていること、を特徴とする請求項 1 1 に記載のエンコード。

【請求項 1 3】

それぞれが、ベース層 ID フィールド (1 0 8) と拡張層 ID フィールド (1 1 2) とで構成された層識別構文要素構造を含む一連のパケット (1 0 6) から成る多層ビデオ信号 (1 0 4) にビデオ信号を符号化するための方法であって、前記方法は、

複数のパケットを含む前記多層ビデオ信号の予め決められた部分 (1 1 3) について、前記多層ビデオ信号の前記予め決められた部分 (1 1 3) に含まれる前記パケットの拡張層 ID フィールド (1 1 2) の仮の最大値を示す最大構文要素 (1 1 4) を前記多層ビデオ信号に挿入し、

前記仮の最大値に基づいて、前記多層ビデオ信号の前記予め決められた部分内の層の最大数を決定し、そして、

前記多層ビデオ信号の層特徴付け構文部分 (1 2 4) を層の最大数と等しい回数繰り返して書き込むことによって、前記多層ビデオ信号 (1 0 4) の最大数の層のそれぞれについて最低 1 つの特徴を信号で伝え、

前記最大構文要素 (1 1 4) によって、前記ビットストリームの中の前記拡張層 ID フィールド (1 1 2) のビット長さが信号で伝えられることを特徴とする方法。

【請求項 1 4】

プログラム符号がコンピュータ上で実行されると、前記コンピュータが請求項 1 0 または請求項 1 3 の方法を実行する、前記プログラム符号を有するコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

本発明は、例えば多層ビデオ符号化の分野などの多層符号化における層特徴の信号に関する。

【背景技術】

【0 0 0 2】

H E V C 基本仕様 [非特許文献 1] などのビデオ符号器規格、および、多ビュー (v i e w) 拡張 [非特許文献 2、3] などのその拡張は、通常、層にされた符号化シナリオにおいて、ビデオデータの個々の符号化された大きな塊が属する層を信号で伝えるためのメカニズムを引き継ぐ。符号化メカニズムはすぐに利用できることを必要とする (図 1 の表に与えられた H E V C の N A L ユニットヘッダー (n u h _ l a y e r _ i d) の中の層

10

20

30

40

50

識別子と比較してください)。

【0003】

ビットについてその有限サイズに関して、個々のHEVC構文要素は、スケーラブル層、ビューおよび/または奥行きを識別するために用いられる64個の値を許容する。例えば、現在のHEVCNALユニットヘッダー構文を使っている多ビュービデオに対して、これは、最大64個のビュー、または、32個のビュー+奥行きの組み合わせの制限を暗示している。これは多くの共通のステレオビューシナリオおよび他の応用のために十分であるけれども、光分野画像処理[非特許文献4、5]またはホログラフィックディスプレイ[非特許文献6]のような応用のための100個を超えるビューを有する莫大なカメラ配列は、層識別子のより大きい空間を信号で伝えるために拡張可能なメカニズムを必要とする。

10

【0004】

さらに、それらの符号化依存(coding dependencies)もしくは互いの空間関係の観点から、所定の層またはビューをグループ化することによって層識別子の空間を構造化することが有利である。さらに、そのような情報を符号化されたビデオビットストリーム内の既に利用可能な情報から引き出すことが耐えられない計算処理源を必要とするので、構造化された層識別子空間の中の符号化依存のコンパクトな表現も利点がある。

【先行技術文献】

【非特許文献】

20

【0005】

【非特許文献1】B.プロス、W.-J.ハン、J.-R.オーム、G.J.サリバン、T.ウィーガント(編集)、「高能率ビデオ符号化(HEVC)テキスト仕様書草稿10」、JCTVC-L1003、ジュネーブ、CH、2013年1月

【非特許文献2】G.テク、K.ウェグナー、Y.チェン、M.ハンニュクセラ、J.ボイス(編集)、「MV-HEVC草稿テキスト3(ISO/IEC 23008-2 P D A M 2)」、JCT3V-C1004、ジュネーブ、CH、2013年1月

【非特許文献3】G.テク、K.ウェグナー、Y.チェン、S.イア(編集)、「3D-HEVCテストモデル説明書、草稿仕様書」、JCT3V-C1005、ジュネーブ、CH、2013年1月

30

【非特許文献4】ウィルバーン、ベネット他、大きいカメラ配列を使用している高性能画像、グラフィックのACMトランスアクション、2005、24。Jg.、Nr. 3、S. 765-776

【非特許文献5】ウィルバーン、ベネットS.他、光分野ビデオカメラ、電子画像処理2002、光学と光通信学のための国際協会、2001。S. 29-36

【非特許文献6】ホリマイ ヒデヨシ他、360度水平線視角を持つフルカラー3Dディスプレイシステム、Proc. Int. 3Dとコンテンツのシンポジウム、2010、S. 7-10

【発明の概要】

【発明が解決しようとする課題】

40

【0006】

従って、例えば、減少したサイド情報オーバーヘッド、および/または、発展したバックワード互換性について、より効率的である多層符号化に関連して多層符号化における層特徴の信号のための概念を提供することが、本応用の目的である。この目的は保留中の独立請求項の主題によって達成される。

【課題を解決するための手段】

【0007】

本応用の面のうちの1つは、多層ビデオ信号の層のための最低1つの特徴の信号(例えば個々の層に対してそれぞれの層が相互層予測を介して直接に関連する依存層の標示など)の信号、または、前述の第2相互依存構文構造の信号に関係している。この面に従って

50

、最大の構文要素は、多層ビデオ信号のパケットの拡張層IDフィールドの最大限に使われた値を示すために、多層ビデオ信号内で信号が伝えられる。例えば、最大の構文要素の範囲は、多層ビデオ信号のいくつかの部分の横切って拡張している、多層ビデオ信号の予め決められた部分である。従って、この面によると、多層ビデオ信号の相対的に大きい予め決められた部分に対して、拡張層IDフィールドによって信号化できる可能な値の可能なドメインの実際に消費された部分についての知識を得ることは、多層ビデオ信号を受信するデコーダやネットワーク要素などの装置に対して適当である。最低1つの特徴が、ベース層IDフィールド値と拡張層IDフィールド値とのそれぞれの結合に対して信号で伝えられる必要はないけれども、むしろ、最大限に仮定された値に基づいて決定された層の最大値のための最低1つの特徴を信号で伝えることは重要である。従って、最低1つの特徴は、層IDを持つ層のために送信/信号することは必要でない。その拡張層IDは、多層ビデオ信号の予め決められた部分内で起こらない。これを越えて、別の実施の形態に従って、最大限に仮定された値の知識は、個々の部分の層IDを信号で伝えるためのサイド情報オーバーヘッドを減らすために、すなわち、多層ビデオ信号のパケット内の拡張層IDフィールドを信号で伝えるために必要なビットを減らすために使用される。

10

【0008】

本応用の面のうちの1つは、多層データストリームの層間の相互層依存の信号化に關係している。この面によると、一方の相互層依存の潜在的な多様性の非常に集中的な制限と、他方の相互層依存の非常に複雑な信号との間の良好な妥協案は、ベース層IDによって表現可能な異なる値のペア間の相互依存を標示している第1相互依存構文構造と、拡張層IDによって表現可能な異なる値のペア間の相互依存を標示している第2相互依存構文構造との方法によって、相互層依存を説明することによって見付けられる。多層データストリームの部分は、層に索引を付けているベース層IDおよび拡張層IDに関連する。この概念に従って、強調は、一方の信号化できる相互層依存の増大した多様性と、他方の相互層依存を信号で伝えるための減少したサイド情報オーバーヘッドとの間を移行する。例えば、共通のベース層IDを持つ層のセットを、それぞれ「クラスタ(群れ)」と呼ぶ。同じ第2相互依存構文構造は、第1の相互依存構文構造を介して、全てのクラスタ内の、および、互いに関連した全てのクラスタ間の相互依存を調節するために、別々に使用される。代わりに、第2相互依存構文構造の2つの例示が、一方のクラスタ内の層と他方の異なるクラスタの層間との相互依存を説明するために用いられる。増大した多様性または減少したサイド情報オーバーヘッドに置かれた強調とは無関係に、相互依存信号概念が、信号オーバーヘッドを低く保持することを生じる。

20

30

【0009】

本応用の面のうちの1つは、多層ビデオ信号のパケットの各々が関連する層IDの信号に關係している。特にこの面は、この層関係を信号で伝える効率的な方法を達成する。それでも、ベース層IDフィールドの所定の値が、ベース層IDフィールドのベース層ID値が0のように拡張できないと制限される符号でバックワード互換性を維持する。この拡張できないベース層ID値についてこの制限を特に回避する代わりに、多層データストリームの部分の層IDは、ベース層IDフィールドを第1副フィールドおよび第2副フィールドに区分することによって拡張できる方法で信号で伝えられる。ベース層IDフィールドの第1の副フィールドが予め決められた基準を満たせば、いつでも、拡張層IDフィールドが提供される。仮にベース層IDフィールドの第1副フィールドが、予め決められた基準を満たさないならば、拡張層IDフィールドは省略される。前記の拡張できないベース層ID値は、ベース層IDフィールドの第1副フィールドが予め決められた基準を満たさないベース層ID値のグループ内に「隠されて」いる。従って、この拡張できないベース層ID値は別々に処理さるのではなく、前者のグループの部分が処理される。むしろ、仮にベース層IDフィールドの第1副フィールドが、予め決められた基準を満たすならば、拡張値は、拡張値のドメインの第1副セットの中に存在するように、多層データストリーム内の信号で伝えられた拡張層IDフィールドから引き出される。そして、仮にベース層IDフィールドの第1副フィールドが、予め決められた基準を満たさないならば、この

40

50

拡張値は、拡張値のドメインの第1副セットに解体する値に設定される。個々の部分が関連する層は、その時、ベース層IDフィールドの第2副フィールドから引き出されるクラスタ値と同様の拡張値を使って索引を付けられる。大体、信号効率は、バックワード互換性の維持にも関わらず失われない。

【図面の簡単な説明】

【0010】

概説された面は結合され又は個々に使用され、上で概説された面の好ましい実例は、従属請求項の主題であり、図に関して後で議論される。

【0011】

【図1】図1は、HEVCのNALユニットヘッダーの構文を示す。

10

【図2】図2は、多層ビデオエンコーダ、ネットワーク要素および多層ビデオデコーダを含む環境を模式的に説明する。多層ビデオエンコーダ、ネットワーク要素および多層ビデオデコーダは、以下の図面において説明された概念のいずれか、または、いずれかの結合によって改善される。

【図3】図3は、ネットワーク装置、および、ネットワーク装置に到達する多層データストリーム内のクラスタな方法の層相互依存において信号で伝える概念を模式的に示す。

【図4】図4は、図3の概念に従って相互層依存を信号で伝える方法を模式的に説明する。

【図5】図5は、図3の信号概念を使う場合に相互層依存を引き出すために、より具体的な例を模式的に説明する。

20

【図6】図6は、層ID拡張を示している構文要素の方法によって拡張されたHEVC構文の一部分を示すことによって、スライスセグメントヘッダーのHEVC構文の可能な拡張を示す。

【図7】図7は、図3の信号概念を実施するために、例示的に拡張されたVPS構文の一例を示す。

【図8】図8は、図7の例の代案を示す。

【図9】図9は、図7の例の代案を示す。

【図10】図10は、図7の例の代案を示す。

【図11】図11は、拡張層IDフィールドの最大限に仮定された値が、データストリームの中で信号によって伝えられる本応用のより一層の概念を設定するための装置を模式的に示す。

30

【図12】図12は、図11の概念を模式的に説明する。

【図13】図13は、拡張層IDフィールドを含むように拡張されたHEVCのスライスセグメントヘッダーを取り出す部分の一例を示す。

【図14】図14は、図13の拡張層IDフィールドの長さを示すために、構文要素を含むように拡張されたVPS構文の一例を示す。

【図15】図15は、拡張層IDフィールドの最大値を示している構文要素を含むように拡張されたVPS構文の一例を示す。

【図16a】図16aは、拡張層IDフィールドを含むように拡張されたSEI拡張の構文構造を取り出す部分の一例を示す。

40

【図16b】図16bは、拡張層IDフィールドを含むように拡張されたSPS拡張の構文構造を取り出す部分の一例を示す。

【図16c】図16cは、拡張層IDフィールドを含むように拡張されたPPS拡張の構文構造を取り出す部分の一例を示す。

【図17】図17は、図11および図12について説明された最大限に仮定された値に依存している構文部分を、複数回繰り返して通ることによって、ビデオデータストリーム内の層に情報を信号で伝えているVPS構文の一例を示す。

【図18】図18は、特定の例に従って、特に、拡張層IDフィールドを示すためにlayer_id_extを使用して、ベース層IDフィールドを示すためにnuh_layer_idを使用して、及び、層IDを示すためにlayer_idを使用して、拡張層I

50

Dフィールド、ベース層IDフィールドおよび層IDの間の模式的な例示関係を示す。そして、拡張層IDフィールドの表現可能な値の数が、`max_layer_id_ext (plus 1)`によって示されている。ここでは、例として、`LengthOfExtension`を2と等しく設定することによって、拡張層IDフィールドの長さを2に選択することによって3に設定されている。

【図19】図19は、拡張層IDフィールドを含むように拡張されたスライスセグメントヘッダーを取り出す部分の一例を示す。

【図20a】図20aは、ベース層IDフィールドの副フィールドの長さを示す構文要素を含むように拡張されたVPS構文の一例を示す。

【図20b】図20bは、拡張層IDフィールドを条件付きで明示的に信号で伝えるように修正されたスライスセグメントヘッダー構文を取り出す部分の一例を示す。

【図21】図21は、図18の概念に従って実施された関係を使用して層IDを引き出しているネットワーク装置の機能を説明するフローチャートを示す。

【発明を実施するための形態】

【0012】

本応用の特定の実施の形態を説明する前に、例示的な多層ビデオ符号器の環境が図2に関して説明される。以下で概説される実施の形態と面は有利に使用される。例示的な多層ビデオ符号器の環境のこの概要は、さらに以下で概説される実施の形態を限定するものと理解してはならないけれども、以下の図について例示的に提供される詳細と機能は、より詳細な実施の形態を生じるように以下でさらに概説される実施の形態の可能な拡張の予定を説明するとみなされる。ともかく、以下に提供された説明の多層ビデオ符号器の環境は、以下で概説される実施の形態の利点の理解をより容易にする。

【0013】

図2は、入力でビデオ素材202を受信する多層ビデオエンコーダ200を説明的に示す。説明の目的だけのために、ビデオ素材202は、ビデオの個々の画像のためのテクスチャや奥行きなどの複数のビューのそれぞれのためにビデオデータを移送する多ビュービデオ信号として説明される。一般的に、ビデオ素材202は1つの場面の多次元サンプリングを表している。1つの次元は時間tである。すなわち、ビデオ素材202はこの場面を時間的にサンプリングし、従って、1つの時間瞬間(`time instant`)当たり1つの画像を生じる。さらに、この場面は異なる視点から捕らえられ、従って、1つのビュー当たり1つのビデオ、または、時間瞬間ごとに1つのビュー当たり1つの画像を生じる。さらに、ビデオ素材202は、場面のテクスチャの空間的サンプリング(すなわち、個々のビュー毎の、場面の色や場面の奥行きマップの空間的サンプリング)に加えて、個々のビュー、または個々のビューの副セットを提供する。図2は、例えば、ハッチングを使用して説明された奥行きマップとハッチングが描かれていないテクスチャマップまたはイメージとの間で異なる。しかし、さらに以下で概説される実施の形態のどれもが、そのような多ビュー素材に限定されないこと注意すべきである。むしろ、図2の説明は、以下で前面に出された説明を緩和するように提供されただけである。

【0014】

多層ビデオエンコーダ200は、多層データストリームまたはビデオ信号204の中にビデオ素材202を符号化するように構成されている。特に、多層ビデオエンコーダ200は、相互層予測を使用して、情報量の異なるレベルと一致する異なる層で、多層データストリーム204の中にビデオ素材を符号化する。これは以下を意味している。多層ビデオエンコーダ200は、複数のパケット206を含むように多層データストリームを生成する。パケット206のそれぞれは、異なる層のうちの1つと関連する。多層ビデオエンコーダ200は、例えば、所定のベース層のパケット206の中に、ビデオ素材202(例えば単に1つのビューのテクスチャまたは同類など)の情報量のあるベースレベルを符号化する。例えば、様々な層のうちのいずれかにパケット206の関連が、個々のパケット206内の層識別構文要素構造208から引き出せる。例えば、層識別構文要素構造208は、個々のパケット206のヘッダー210内に含まれている。例えば、説明の目

10

20

30

40

50

的のために、数字 1、2、および 3 が、パケット 206 の中に記入されるように示される。従って、層 ID を持つ層がそれぞれ 1、2、および 3 によって説明される。例えば、これらの層の間で、層 ID 1 を持つ層は、情報量のベースレベルを提供する最も低い層またはベース層を形成する。他の層のパケット 206 は、1 以上または様々な情報タイプについてのみ、ベース層のパケット 206 で利用可能な情報量を増大する。例えば、特定の層のパケット 206 は、ベース層またはリファレンス層のパケット 206 内で既に符号化されたビューに加えて、その中で符号化された別のビューを持つ。代わりに、特定の層のパケット 206 は、ビデオ素材 202 の場面の奥行き情報（ビューの奥行き情報など）を持ってよい。そのテクスチャは、ベース層またはより一般的にはより下の層であるリファレンス層のパケットの中に既に符号化されている。同様に、表面反射率情報は、別の層（すなわち、照明状況とは無関係にその場面内のオブジェクトの表面反射率についての場面の空間サンプリング）のパケットの中に符号化される。そして、さらに、アルファ混合情報は、別の層（すなわち、個々の画像サンプルと一致している場面ポイントの透明度の空間のサンプリング）のパケットの中に符号化される。また、特定の層のパケット 206 は、特定の色組み合わせ情報を追加し、または、空間解像度を増大する（すなわち、空間解像度改良を提供する）。同様に、特定の層のパケット 206 は、SNR 解像度改良を単に提供する、すなわち、符号化されたビデオ素材の信号対ノイズ比を増大させる。様々な層が関係している限り、多層データストリーム 204 内の冗長性を避けるために、相互層予測が多層ビデオエンコーダ 200 によって使用される。すなわち、特定の層のパケット 206 は、前者の層についてリファレンス層と呼ばれる 1 つ以上の他の層から得られるように、相互層予測について予測残差をその中で符号化する。相互層予測は、ビデオ素材 202 を説明している構文要素やパラメータなどの異なる実体に適用される。例えば、特定の層のパケット 206 によって運ばれた予測残差は、1 つ毎の画素に基づいて空間ドメインの中の 1 つ以上のリファレンス層から得られるように相互層予測を訂正する。代わりに、多層ビデオエンコーダ 200 は、例えば、変換残差符号化を使用し、相互層予測訂正が変換係数ベース毎に DCT ドメインなどの変換ドメイン内で起こる。代わりに又は追加的に、多層ビデオエンコーダ 200 は、空間的および / または時間的な予測を使用するハイブリッドビデオ符号器タイプであり、例えば、相互層予測は、動作ベクトルの改良に追加的または代替的に関係する。さらに、多層ビデオエンコーダ 200 は、異なる予測モード（空間的および時間的予測など）が適用されるユニットの中の符号化ブロックの中に、および / または、前記の変換が予測残差上で実行されるユニットの中の変換ブロックの中に、および / または、特定の符号化パラメータがビデオ素材 202 を符号化するためにセットされるユニットの中の別のブロックの中に、ビデオ素材 202 の画像を副分割するために、階層的な多ツリー副分割を使用する。そして、改めて別の層のための副分割情報を信号で伝える代わりに、副分割化が 1 つ以上のベース層のいずれかから完全に採用されるか、または、その副分割情報についての改良が信号で伝えられるかのいずれかである。

【0015】

層間の相互層依存の方法によって、多層ビデオ信号 204 の層が、ツリー 212 の枝を介して相互に関係付けられる。ツリー 212 の節（ノード）は前記の層によって形成される。層間の相互層依存についての情報を移送する（すなわち、復号化側で利用できる相互依存ツリー 212 上の情報を提出する）ために、多層ビデオ信号 204 は、そこに符号化されたその情報を持つ。

【0016】

図 2 において、例えば、多層ビデオエンコーダ 200 が、ビデオデータストリーム 204 の中に、情報 214 を構成する又は移送するハイレベル構文パケット 216 を散在させることが説明されている。情報 214 は、ツリー 212 の中の相互層依存を説明するために相互依存構文構造を含む。

【0017】

しかし、ツリー 212 の情報は、エンコーダおよびデコーダでの欠陥（デフォルト）によって、代替的に知られている又は再建可能である。従って、追加的または代わりに、層

10

20

30

40

50

関連情報 2 1 4 は、ある層 I D 拡張機構に関連した情報を含む。特に、多くの応用に対して、層の適度な数は、多層ビデオ信号 2 0 4 を形成するために重要であるけれども、いくつかの別の応用は、層識別構文要素構造 2 0 8 によって信号化可能な非常に小さい数の層から損害を被る。すなわち、層識別構文要素構造 2 0 8 が、高い数の層を必要とするそれらの応用に対して、莫大な数の層に適応するように、層識別構文要素構造 2 0 8 を構成することは、応用の全体の多様性を考慮するとき、高いサイドの情報オーバーヘッドの中に不利を生じさせる。そして、層識別構文要素構造 2 0 8 が適度な数の層にのみ適応するように、層識別構文要素 2 0 8 を構成することは、データストリーム 2 0 4 を根底にしている多層ビデオ符号器によって十分にサポートされることから別の応用を除外する。従って、ある拡張機構は、データストリーム 2 0 4 のモード間で切り替えできるように使用される。ここで、層識別構文要素構造 2 0 8 はより低い適数の層にのみ適応する。そして、層識別構文要素構造が高い数の層および情報 2 1 4 に適応するモードは、これらのモード間の切り替えに追加的または代替的に関与する。

10

【 0 0 1 8 】

ハイレベル構文パケット 2 1 6 の妥当性の範囲は、多層ビデオ信号 2 0 4 の全体を取り巻くけれども、例えば、それらは多層ビデオ信号 2 0 4 が時間的に分割される大きな塊と関連するように、パケット 2 1 6 が属する予め決められた部分がより小さいことも可能である。それぞれの大きな塊は画像のシーケンスである。例えば、D A S H または別の適したストリーミングプロトコルを使用して、多層ビデオエンコーダ 2 0 0 は、ちょうど言及した予め決められた部分のユニットの中のビットストリーム 2 0 4 の中に符号化された層数、ツリー構造 2 1 2 の形式の相互層依存、および / または、層識別構文要素構造 2 0 8 について拡張または不拡張の間の切り替えを変更する。とにかく、ハイレベル構文パケット 2 1 6 を周期的に送信することは、受信者を、ランダムアクセスに基づいて、両者間で多層ビデオ信号 2 1 4 を復号化し始めるために複数のランダムアクセス時間瞬間を持つことができるようにする。

20

【 0 0 1 9 】

図 2 は、多層データストリームを受信するための装置も例示的に示す。例えば、オプションのネットワーク要素 2 1 8 は、多層ビデオデータストリーム 2 0 4 を受信し、それを多層ビデオデータストリーム 2 0 4 の受信者に転送するために処理する。従って、図 2 は、多層ビデオデータストリーム 2 0 4 を復号するように構成された多層ビデオデコーダ 2 2 0 も示す。両方の実体、すなわちネットワーク要素 2 1 8 および多層ビデオデコーダ 2 2 0 は、多層データストリーム 2 0 4 を処理するための装置のための例を形成する。

30

【 0 0 2 0 】

ネットワーク要素 2 1 8 は、そこから様々な層間の相互層予測を使って、多層データストリーム 2 0 4 を復号できる必要はない。これにもかかわらず、ネットワーク要素 2 1 8 および多層ビデオデコーダ 2 2 0 (すなわち受信者) は、ツリー 2 1 2 によって定義されるように、様々な層とそれらの層間の相互層依存とに関連するパケット 2 0 6 を知らせる必要がある。ネットワーク要素 2 1 8 は、例えば、提供された追加の情報量がネットワーク要素 2 1 8 と多層ビデオデコーダ 2 2 0 との間の別の連結内でそれぞれ予備的に必要ではない層に関連したデータストリーム 2 0 4 のパケット 2 0 6 を除去する。例えば、連結内のビットレート欠点のために、追加の情報量または同類を再生するための受信者の無能力を除去する。同様に、多層ビデオデコーダ 2 2 0 も、現在利用できる計算能力、空間解像度などの再生装置のディスプレイ性能、インプットビューまたは同類の最大数のような外部パラメータに対応した特定の層のパケット 2 0 6 のうちのいくつかを捨てることを決める。すなわち、装置 2 1 8 / 2 2 0 は、パケット 2 0 6 の層識別構文要素構造 2 0 8 を読み取り、もし存在するならば、層間の相互層依存を情報 2 1 4 から引き出し、および / または、情報 2 1 4 に対応した層識別構文要素構造 2 0 8 に関する拡張モードまたは不拡張モードの間で切り替え、および / または、情報 2 1 4 から層に関する別の特徴を読み取ることが可能である。

40

【 0 0 2 1 】

50

それを超えて、多層ビデオデコーダ 220 は、その層と、この層が相互予測依存によって相互に関連した全ての層とに関連したパケット 206 の情報を収集して使用することによって、所定のレベルに引き上げて、インバウンド（帰航）データストリーム 204 からビデオ素材 202 を再構成することができる（ツリー構造 212 を参照）。すなわち、上で概説したように多層ビデオデコーダ 220 は、多層ビデオデコーダ 220 がその層の 1 つ以上のリファレンス層のパケット 206 から引き出す相互層予測に関する予測残差として、特定の層のパケット 206 を使用する。この点について、多層ビデオデコーダ 220 は、上で概説したように、変換残差復号化、ハイブリッドビデオ復号化、階層的な多ツリー分割化および/または別の符号化概念を使用するデコーダである。

【0022】

10

完全性だけのために、図 2 は、多層ビデオエンコーダ 200 とデータストリーム 204 が、異なる時間瞬間または画像に属しているパケット 206 が互いに挟まれない方法で、時間瞬間毎に連続的なアクセスユニット 222 を形成するように、パケット 206 がデータストリーム 204 内に含まれるように例示的に提供されることを示す。それぞれは、そのアクセスユニット 222 の個々の時間瞬間に属しているパケット 206 を収集している。その結果、別の時間瞬間に属しているパケット 206 は、データストリーム 204 内において、このアクセスユニット 222 の前又は後に続くかのいずれかである。しかし、この制限は説明目的のために選択されただけであり、データストリーム 204 内のパケット 206 のより緩和した配置が代わりに選択されてもよいことは注意すべきである。

【0023】

20

以下において、ツリー構造 212 に従って相互層依存を信号で伝える可能性が、より詳細に説明される。前記したように、この信号は情報 214 を含む。この可能性に従って、層はクラスタにグループ化される。データストリーム内において、一方のクラスタ間の相互層依存と他方のクラスタ内の層とが、別々に信号で伝えられる。その限りにおいて、以下で前面に出される説明は、相互層依存 212 を信号で伝えるために、図 2 の情報 214 を実施する可能性を表現している。しかし、既に上で注意したように、図 3 に関して以下に説明される詳細は、図 2 に発表された詳細に限定されない。むしろ図 2 は図 3 に関する説明のための可能な実施貯蔵所と見られるべきである。

【0024】

図 3 は、図 2 のネットワーク要素 218 または多層ビデオデコーダ 220 であるネットワーク装置 18 を説明する。そして、多層データストリーム 10 は、図 2 のデータストリーム 204 のように処理するように構成される。

30

【0025】

図 3 は、層 ID 1 ID によって識別された異なる層で、ビデオ素材をその中に符号化された多層データストリーム 10 を説明する。個々の層は、情報量の異なるレベルと一致する。この点における可能性は、既に図 2 に関して議論されているけれども、より容易な理解のために、特定のビューをデータストリーム 10 に追加している個々の層について考える。しかし、これは図 3 の説明を制限することを意図しない。個々の層は、代替的に、情報量（例えば、ビューの数および空間解像度の数または同類など）のための異なる手段の結合に一致している。

40

【0026】

従って、多層データストリーム 10 は、図 2 のパケット 206 に例示的に一致している複数のパケット 12 から成る。そのようなパケット 12（または 206）は、例えばビデオ信号 10 の中に符号化された個々の画像の波面並列処理を許している副ストリームである。副ストリームは、順に、NAL ユニット、スライスまたは同類などのより小さいユニットから構成されている。しかし、図 2 および図 3 のパケット 12 またはパケット 206 は、NAL ユニット、スライスまたは別のユニットでもあることが、同封して提出される。

【0027】

個々のパケット 12 は、異なる層のうちの 1 つと関連する。そして、多層データストリ

50

ーム10のビット消費を減らすために、個々のパケット12がデータストリーム10のより低い層の特定の副セットのパケットに「残差」を単に追加するように、相互層予測が使用される。符号14で示されるように、「より低い層」は個々の層毎に小さいドットによって説明される。

【0028】

図3において示された実施の形態に従って、相互層予測依存は、以下でより詳細に説明され、図において実線16を使用して説明されるように限定される。特に、層IDは、以下において説明されるように、2つの値の連結によって形成される。特に、多層データストリーム10を受信するネットワーク装置は、パケット12毎に、図2のリファレンス記号208を使用して示された層ID構文構造を読み取る。しかし、ここは、ベース層IDフィールド20と、条件付きで、-例えば拡張機構をオンとオフに切り換えるベース層IDフィールドまたはハイレベル構文要素に条件的に依存する-拡張層IDフィールド22とから成る。例示的に、両方のフィールドを持っているパケット12のみが、図において示される。しかし、前記したように、ベース層IDフィールド20の1つ以上の値が、個々のパケット12のためのフィールド22の不存在を信号で伝える。例えば、拡張機能全体は、データストリームの中の拡張フラグを介して、データストリームの中でオン/オフ切り換え可能である。拡張層IDフィールドを必要とする又は必要としない可能な条件は、前記したように、ベース層IDフィールドが0である、または、特定のビットを設定する又は設定しない、または、フィールド20の副部分内の特定の値を仮定する、または、ベース層IDフィールドがある値より大きい又は小さい値を持っていることである。別の例が以下に発表される。

【0029】

層ID構文構造20と22に基づいて、ネットワーク装置18は、個々のパケット12が関連する層を識別している層ID、すなわちLIDを引き出す。異なる可能性は以下に説明される。ネットワーク装置18は、多層データストリーム10から、例えば前記で識別された`direct_dependency_flags`を含んでいる第1相互依存構文構造、および、例えば以下で示された`direct_ext_dependency_flag`および/または`general_direct_ext_dependency_flag`を含んでいる第2相互依存構文構造も読み取る。第1相互依存構文構造は、ベース層IDフィールド20によって表現可能な異なる値のペア間の相互依存を2進数の方法で示す。これに対して、第2相互依存構文構造は、拡張層IDフィールドによって表現可能な異なる値のペア間の相互依存を2進数の方法で示す。両方の相互依存構文構造に基づいて、ネットワーク装置18は、その時、異なる層(図において符号14で描かれたものなど)の間の相互層予測依存を明らかにしている相互層依存マトリクスを形成する。派生される方法は、以下において、第1および第2相互依存構文構造に関するフォアネクストループを使用している擬似符号を使用して説明され、そして、以下の図に関しても説明される。しかし、信号で伝えられた相互層予測依存が、実際にデータストリームにおいて使われることは必要ないことは注意すべきである。むしろ、可能な相互層予測依存の信号化は、デコードまたは別のネットワーク装置に、相互に関係付けられたパケット、すなわち、リファレンスパケットに先がけて相互層予測依存に従って別のパケットによってリファレンスされたパケットが、適切な順序で利用可能である必要なステップを取ることを指示するためにある。

【0030】

以下の説明から明らかになるように、相互層依存マトリクス14の構築は、第2相互依存構文構造が、相互層依存マトリクス14がベース層IDフィールド20内の同じ値から引き出され、その結果、それと関連している層IDの層間の相互層予測依存に関連する全ての事例に適用されるように実行される。また、以下に、より詳細に説明された別の実施の形態に従って、第2相互依存構文構造が、例えばベース層IDフィールド20の個々の可能な値に対して、または、ベース層IDフィールド20の可能な値の副セットに対して読み取られ、そして、ビットストリーム10内に数回送信される。一方、第2相互依存構

文構造の様々な具体例が、索引付けを使用して、例えば、以下に示された例の中の `included_nuh_layer_id` を使用するなどして、ベースの層 ID フィールド 20 の可能な値に関係付けられる。別の実施の形態においてさえ、構造は、全ての可能な相互層予測依存が許されて、第 2 相互依存構文構造の方法で、すなわち第 1 相互依存構文構造が相互依存の存在を示すベース層 ID フィールドの異なる値のペア毎に第 2 相互依存構文構造を送信することによって描写できることにおいて一般化される。

【0031】

言い換えると、図 4 においてさらに説明されるように、図 3 は、別々の相互依存構文構造 24 と 26 が、データストリームの個々のパケット内のフィールド 20 と 22 によって提供されたベース層 ID および拡張層 ID の範囲の間のインタフェースと一致している両方の相互依存構文構造 24 と 26 の範囲の間のインタフェースを持つ階層的な方法で、相互層依存を説明するために使用されることに従って、相互層依存を信号で伝えるための例を示す。フィールド 20 と 22 によって提供されるようなベース層 ID および拡張層 ID は、フィールド 20 と 22 が構成される個々のパケットの層 ID を独自に定義する。拡張層 ID およびベース層 ID の結合によって表現可能な全ての表現可能な層 ID のセットは、円 28 内でドット 30 によって示される。すなわち、個々のドット 30 は、ベース層 ID および拡張層 ID の異なるカップルに一致している。例えば、層 ID は、ベース層 ID および拡張層 ID の連結である。フィールド 20 だけによって提供されるベース層 ID を使用して、層 ID 30 の完全なセット 28 が、以下のクラスタ 32 と称される層 ID の解体セット 32 に分割される。特定のクラスタ 32 に属している全ての層 ID は、同じベース層 ID を持つ。以前に図 2 に関して説明したように、層 ID 30 と関連した層が、ドット 30 の間の点線を使用して、図 4 において説明したこれらの相互層依存を持つ相互層予測のためにツリー似の方法で相互に連結される。理解し易くするために、実際の相互層依存の副セットだけが図 4 において説明される。

【0032】

ともかく、2つのクラスタ 32 の間の連結化（このペアの第 1 クラスタ 32 の層と第 2 クラスタ 32 の層との間の相互層依存の方法による）が、第 1 相互依存構文構造 24 の方法によって示される。すなわち、第 1 相互依存構文構造 24 は、層間の相互依存を粗く又はクラスタ状（群れ状）に説明する。図 4 おいて、これらの相互依存は、クラスタ 32 の間の実線を使って説明される。第 1 クラスタの最低 1 つの層が第 2 クラスタの中の 1 つのクラスタに連結している第 1 および第 2 クラスタ 32 の全てのペアは、相互連結され、第 1 相互依存構文構造 24 において表示されている。第 2 相互依存構文構造 26 は、その時、第 1 相互依存構文構造 24 によって相互連結されると表示されたクラスタ 32 のペアの層のどちらが、相互層予測によって互いに実際に関連するかを明確にする。すなわち、第 2 相互依存構文構造 26 は、微細粒状の相互依存を明確にする。しかし、第 2 相互依存構文構造 26 は、個々のクラスタ 32 内の層の間の相互依存、すなわち層間の内部クラスタ依存も個別に定義する。図 4 において、例えば 6 つのクラスタ 32 が示され、従って、第 2 相互依存構文構造 26 が別のクラスタ 32 の層間の微細粒状の相互依存を調節するクラスタ 32 の 15 個の可能なペアに加えて、第 2 相互依存構文構造 26 が相互依存を内部で調節する 6 つのクラスタを生じる。上で概説したように、そして、以下に更に概説するように、従って、最大で第 2 相互依存構文構造の 1 個 ~ 21 個の実例 34 が、つまり、個別にクラスタ当たり 1 つとクラスタ 32 の連結されたペア当たり 1 つとが存在する。明らかに、第 2 相互依存構文構造 26 は、第 1 相互依存構文構造 24 によって連結されないように信号で伝えられたクラスタ 32 のペアに対して、信号で伝えられる必要はなく、従って、貴重なサイド情報ビットレートが抑制される。クラスタ 32 の異なるペアの層の相互層依存を説明するために第 2 相互依存構文構造 26 を使う場合において、クラスタ 32 当たりの層数が、全てのクラスタ 32 に対して等しいことに注意するべきである。仮に、層 ID がベース層 ID および拡張層 ID の連結を使って説明されるならば、これは典型的な例である。しかし、クラスタ 32 の層数がクラスタ 32 の間で理論上変わることは注意するべきである。その典型的な例において、第 2 相互依存構文構造 26 の個々の実例 34 が、

例えばデータストリーム内でクラスタ 3 2 の個々の相互連結されたペアに対して信号で伝えられ、最低 1 つの実例 3 4 が個々のクラスタサイズに対して送信される。

【 0 0 3 3 】

図 3 は、例えば、層 ID が、最も重要な数字としてベース層 ID を使い、より重要でない数字として拡張層 ID を使うことによって、ベース層 ID および拡張層 ID から得られる場合を説明する。図 3 は、1 つの実例 3 4 が 1 つのクラスタ内の層の相互層依存を記述するために用いられ、別の実例 3 4 が異なるクラスタの層間の依存を記述するために用いられる例示的な場合も説明する。完全性のために、図 3 のマトリクス 1 4 が層 ID として多くの線と多くの列を持つことが、注意される。対角線の下半分だけが満たされる。なぜなら、単に、どの層も、相互層予測の方法で、前の層に（すなわち階層的に低い層に）依存するからである。図 3 の例において、列数は、相互層予測の使用によって別の層（すなわちベース層）に依存する層の層 ID と一致している。これらのベース層は、例えば 2 進数によって示される。2 進数のゼロが、個々の層を予測している相互層に関与していない層を示す一方、その層 ID は現在の列と一致する。その限りにおいて、図 3 の実施の形態において、第 2 相互依存構文構造 2 6 はマトリクス 1 4 の副マトリクスを多少説明する。

10

【 0 0 3 4 】

上に既に記述したように、第 1 相互依存構文構成 2 4 および第 2 相互依存構文構成 2 6 は、ハイレベルパケット 2 1 6 内の情報 2 1 4 によって構成される（図 2 と比較してください）。図 5 は、第 1 相互依存構文構造 2 4 が符号 3 6 で描かれた層クラスタの間の相互依存を明らかにする例を説明する。例えば、ベース層 ID 2 を持つクラスタは、ベース層 ID 2 および 1 を持つクラスタに依存する。

20

【 0 0 3 5 】

また、第 2 相互依存構文構造の第 1 実例 3 4 は、データストリームの中に存在し、副マトリクス 3 8 の形式で図 5 の中に描かれた層の間の内部クラスタ依存を調節する。さらに、図 5 の例に従って、データストリームは、また、異なるクラスタの層の層状相互依存を調節する第 2 相互依存構文構造 2 6 の実例 3 4 を含む。特に、第 2 実例は、リファレンスされたクラスタのエンハンスメント層 ID 当たり 1 つの行、および、リファレンスしているクラスタのエンハンスメント層 ID 当たり 1 つの列を持っている副マトリクス 4 0 を介して、異なるクラスタの層間の依存を説明する。

【 0 0 3 6 】

30

図 5 の例において、副マトリクス 3 8 は、マトリクス 3 6 がクラスタ間の相互依存を示す個々の位置（すなわち、1 が置かれる個々の位置）に置かれ、そのクラスタは同じベース層 ID のクラスタ（すなわち、マトリクス 3 6 の対角線上にあるクラスタ）である。そして、副マトリクス 4 0 は、マトリクス 3 6 が異なるベース層 ID のクラスタ間の相互依存を「1」で示す位置に置かれる。結果は符号 4 2 で示される。

【 0 0 3 7 】

マトリクス 4 2 などのマトリクスを介する相互層依存の説明が、相互層依存を説明するための単なる 1 つの例であることは注意すべきである。別の説明が同様に使われうる。マトリクス 3 6 ~ 4 0 が第 1 および第 2 相互依存構文構造によって符号化される方法は、次の通りである。第 1 相互依存構文構造 2 4 は、対角線の下でかつ対角線を含むマトリクス 3 6 の各係数に対して、2 進値を信号で伝える。マトリクス 3 8 を標示する第 2 相互依存構文構造 2 6 の実例 3 4 は、対角線の下でかつ対角線を含むマトリクス 3 8 の各係数に対して、2 進値を信号で伝える。マトリクス 4 0 を標示する第 2 相互依存構文構造 2 6 の実例 3 4 は、マトリクス 4 0 の全ての係数に対して、2 進値を信号で伝える。

40

【 0 0 3 8 】

相互層依存をどのように信号で伝えるかについて可能性を説明した後に、より詳細な実例が H E V C 規格の拡張として以下に例示的に提供される。

【 0 0 3 9 】

特に、層識別クラスタ化およびクラスタ依存信号化は、存在する符号器の中に次の通り構築される。

50

【0040】

2つの構文要素 `nuh_layer_id` と `layer_id_ext` が、ビデオビットストリーム内の符号化された層を、いわゆる互いの空間関係などの特性に基づいて依存又は別の物を符号化しているクラスタの中にグループ化するために使用される。クラスタの中への層識別子の構造化は、個々のクラスタ内の等しく構造化された符号化依存を持つクラスタに対して許される。すなわち、定義されたクラスタの全て又は副セット内の符号化依存は、同じである。特定の層のリファレンス層を決めるために、依存フラグの追加のセットとしてクラスタ内の依存を信号で伝えること、および、依存を既存の依存信号化（HEVC拡張のVPS拡張の中の `direct_dependency_flag` と比較してください）に結合することは、図6に記述されるように行われる。

10

【0041】

0と等しい `direct_ext_dependency_flag[i][j]` は、拡張インデックス（索引）`j` を持つ層が、同じ層クラスタ内の拡張インデックス `i` を持つ層に対して、直接的リファレンス層でないことを規定する。1と等しい `direct_dependency_flag[i][j]` は、拡張インデックス `j` を持つ層が、同じクラスタ内の拡張インデックス `i` を持つ層に対して、直接的リファレンス層であることを規定する。`direct_ext_dependency_flag[i][j]` が、0から $(1 < \text{layer_id_ext_len}) - 1$ の範囲内の `i` と `j` に対して、存在しないとき、それは0に等しいと推定される。

【0042】

20

代替の構文は、未使用の層識別子に依存を符号化することを信号で伝えることを避けるために、`max_layer_id_ext` によって `direct_ext_dependency_flag` 構文要素のループを制限する。

【0043】

クラスタ内の及びクラスタ間の符号化依存のより柔軟な信号は次の通り許される。

- a) クラスタ毎に基づいてクラスタ内の依存を符号化する信号によって。
- b) クラスタ依存セットの数の定義、および、定義されたクラスタ依存セットが適用されるクラスタの識別（例えば、`nuh_layer_id` によってどちらか一方）によって。
- c) クラスタ間の第1依存が信号で伝えられ、2つの依存するクラスタ内の層の第2依存が信号で伝えられる方法の中の依存の階層的信号によって。
- d) b) と c) の組み合わせによって。従って、以下のデータが信号で伝えられる。
 - 1. クラスタ間の依存（例えば、リファレンスのペアおよび依存するクラスタ）。
 - 2. リファレンス（リファレンスされた）クラスタおよび依存する（リファレンスしている）クラスタ内の層間の依存を定義しているクラスタ依存セットの数。
 - 3. 個々のクラスタ依存セットに対して、クラスタ依存が適用する標示（1において信号で伝えられるように）。

30

クラスタ内の層間に依存があるとき、c) と d) に対してクラスタが自身に依存できることに注意しなさい。

a) から d) は、構造化のために使用される特性空間（例えばカメラの空間的位置）が均一に使われないとき、層識別子の空間を構造化することを許す。1つの例が、空間密度が一定でない二次元カメラレイ、または、矩形ではない二次元カメラレイである。以下は、上記に従って、複数の方法において、クラスタ依存セットの定義を信号で伝えるための構文例を与える。

40

【0044】

a) 図7は、クラスタ毎に基づいて、クラスタ内の依存を符号化することを定義するビデオパラメータセット（設定）拡張構文について例示的に実施の形態を与える。

【0045】

0と等しい `direct_dependency_flag[i][j]` は、インデックス `j` を持つ層またはクラスタが、インデックス `i` を持つ層またはクラスタに対して、直

50

接的なリファレンス層またはリファレンスクラスタでないことを規定する。1と等しい `direct_dependency_flag[i][j]` は、インデックス `j` を持つ層またはクラスタが、インデックス `i` を持つ層またはクラスタに対して、直接的なリファレンス層またはリファレンスクラスタであることを規定する。`direct_dependency_flag[i][j]` が、0から `vps_max_layers_minus_1` の範囲内で、`i` と `j` に対して存在しないとき、それは0に等しいと推定される。

0と等しい `general_direct_ext_dependency_flag[i][j]` は、拡張インデックス `j` を持つ層が、同じ層クラスタ内の拡張インデックス `i` を持つ層に対して、直接的なリファレンス層ではないことを規定する。1と等しい `general_direct_ext_dependency_flag[i][j]` は、拡張インデックス `j` を持つ層が、同じクラスタ内の拡張インデックス `i` を持つ層に対して、直接的なリファレンス層であることを規定する。`general_direct_ext_dependency_flag[i][j]` が、0から `max_layer_id_ext` の範囲内で、`i` と `j` に対して存在しないとき、それは0に等しいと推定される。

10

0と等しい `direct_ext_dependency_flag[i][j][k]` は、拡張インデックス `k` を持つ層が、`i` 番目の層クラスタ内の拡張インデックス `j` を持つ層に対して、直接的なリファレンス層ではないことを規定する。1と等しい `direct_ext_dependency_flag[i][j][k]` は、拡張インデックス `k` を持つ層が、`i` 番目の層クラスタ内の拡張インデックス `j` を持つ層に対して、直接的なリファレンス層であることを規定する。`direct_ext_dependency_flag[i][j][k]` が、0から `max_layer_id_ext` の範囲内で、`i`、`j` および `k` に対して存在しないとき、それは0に等しいと推定される。

20

【0046】

b) 図8は、クラスタ依存セットの数の信号 (`number_of_cluster_dependency_sets`) を通して、クラスタ内の依存を符号化すること、個々のクラスタ依存セット内の依存 (`direct_ext_dependency_flag`) を符号化すること、および、クラスタ (たとえば、クラスタの個々の `nuh_layer_id` のためのポジティブなフラグを通して、与えられた構文要素 `included_nuh_layer_id` によって識別されたクラスタ) を適用することを定義するビデオパラメータセット拡張の中の構文について例示的な実施の形態を与える。

30

【0047】

c) 図9は、クラスタ間の第1依存が信号で伝えられ、および、2つの依存するクラスタ内の層の第2依存だけが信号で伝えられる方法で、依存の階層的な信号のための例示的な実施の形態を与える。

【0048】

0と等しい `direct_dependency_flag[i][j]` は、インデックス `j` を持つ層またはクラスタが、インデックス `i` を持つ層またはクラスタに対して、直接的なリファレンス層またはリファレンスクラスタではないことを規定する。1と等しい `direct_dependency_flag[i][j]` は、インデックス `j` を持つ層またはクラスタが、インデックス `i` を持つ層またはリファレンスクラスタに対して、直接的なリファレンス層またはリファレンスクラスタであることを規定する。`direct_dependency_flag[i][j]` が、0から `vps_max_layers_minus_1` の範囲内で、`i` と `j` に対して存在しないとき、それは0に等しいと推定される。

40

注記：仮に、`layer_id_ext_len` が0より大きければ、`direct_dependency_flag[i][j]` は、クラスタ間の依存を信号で伝える。そうでなければ (`layer_id_ext_len` が0と等しいならば)、`direct_dependency_flag[i][j]` は、層のための依存を信号で伝える。

【0049】

0と等しい `direct_ext_dependency_flag[i][j][k]`

50

][*l*]は、*j* 番目のクラスタの中の *l* 番目の層が、*i* 番目のクラスタの中の *k* 番目の層に対して、直接的なリファレンス層ではないことを規定する。1と等しい `direct_ext_dependency_flag[i][j][k][l]` は、*j* 番目のクラスタの中の *l* 番目の層が、*i* 番目のクラスタの中の *k* 番目の層に対して、直接的なリファレンス層であることを規定する。`direct_ext_dependency_flag[i][j][k][l]` が存在しないとき、それは0に等しいと推定される。

【0050】

d) 図10は、b)とc)の組み合わせのための例示的な実施の形態を与える。

【0051】

0と等しい `direct_dependency_flag[i][j]` は、インデックス *j* を持つ層またはクラスタが、インデックス *i* を持つ層またはクラスタに対して、直接的なリファレンス層またはリファレンスクラスタでないことを規定する。1と等しい `direct_dependency_flag[i][j]` は、インデックス *j* を持つ層またはクラスタが、インデックス *i* を持つ層またはリファレンスクラスタに対して、直接的なリファレンス層またはリファレンスクラスタであることを規定する。`direct_dependency_flag[i][j]` が、0から `vps_max_layers_minus1` の範囲内で、*i* と *j* に対して存在しないとき、それは0に等しいと推定される。

注記：仮に、`layer_id_ext_len` が0より大きければ、`direct_dependency_flag[i][j]` は、クラスタ間の依存を信号で伝える。そうでなければ(`layer_id_ext_len` が0と等しいならば)、`direct_dependency_flag[i][j]` は、層のための依存を信号で伝える。

【0052】

0と等しい `direct_ext_dependency_flag[n][k][l]` は、`dependency_set_applies_flag[n][i][j]` が1に等しいとき、*j* 番目のクラスタの中の *l* 番目の層が、*i* 番目のクラスタの中の *k* 番目の層に対して、直接的なリファレンス層ではないことを規定する。1と等しい `direct_ext_dependency_flag[n][k][l]` は、`dependency_set_applies_flag[n][i][j]` が1に等しいとき、*j* 番目のクラスタの中の *l* 番目の層が、*i* 番目のクラスタの中の *k* 番目の層に対して、直接的なリファレンス層であることを規定する。

0と等しい `dependency_set_applies_flag[n][i][j]` は、*i* 番目のクラスタの中の層と *j* 番目のクラスタの中の層との間の依存が、`direct_ext_dependency_flag[n][k][l]` によって規定されないことを規定する。1と等しい `dependency_set_applies_flag[n][i][j]` は、*i* 番目のクラスタの中の層と *j* 番目のクラスタの中の層との間の依存が、`direct_ext_dependency_flag[n][k][l]` によって規定されることを規定する。`dependency_set_applies_flag[n][i][j]` が存在しないとき、それは0に等しいと推定される。`dependency_set_applies_flag[n][i][j]` が、0から(`number_of_cluster_dependency_sets - 1`)の範囲内で、*i* と *j* と全ての *n* との特定の結合に対して0と等しいとき、*j* 番目のクラスタの中の層は、*i* 番目のクラスタの中のどの層に対しても、直接的なリファレンス層でないことを含む。

【0053】

従って、HEVC拡張の中に図2から図5のクラスタに基づいた相互依存信号を実行する上の例において、図2のパケット206はスライスまたはスライスセグメントであり、パケット216は例えばビデオ、画像または連続パラメータセットなどのパラメータセットである。図7の構文例に従って、第1相互依存構文構造24は、*i* *j* を持つクラスタ *i*、*j* の各ペアに対して、構文要素 `direct_dependency_flag` を使

10

20

30

40

50

って信号で伝えられる。ここで、`vps_max_layers_minus1`は、ベース層ID-1の最大値である。従って、構文要素は、例えば図5の粗いマトリクス36を明らかにする。`unique_cluster_dependencies_flag`と称する構文要素44は、2つのオプション（第2相互依存構文構造26の1つの実例は全てのベース層IDに提出されて適用される、または、もう1つの実例は個々のベース層ID i に対して送られるのいずれか）の間を区別する。第1ケースにおいて、フラグ`general_direct_ext_dependency_flag`は、共通して、ベース層ID毎に38に従って副マトリクスを築く。第2ケースにおいて、フラグ`direct_ext_dependency_flag`は、別々に、ベース層ID毎に符号38に従って副マトリクスを築く。`unique_cluster_dependencies_flag`は、両方のオプションの間を切り替える。従って、図7の場合において、仮に、`unique_cluster_dependencies_flag`が0に等しいならば、結果として生じる副マトリクス38は、マトリクス36内の、リファレンスしている及びリファレンスされたクラスタIDに等しいように一致している個々のポジションに置かれる。ここで、1は`direct_dependency_flag`によって示される。第1相互依存構文構造24が異なるクラスタIDのクラスタ間の相互依存を示す残留ポジションは、予め決められた副マトリクス（すなわち、エンコーダ、ネットワーク要素および多層ビデオデコーダなどの全ての参加装置によって知られている副マトリクス）を使用して、欠陥によって満たされる。1に等しい`unique_cluster_dependencies_flag`の場合において、第1相互依存構文構造24が、等しいリファレンスされた及びリファレンスしているクラスタIDのポジションで1を示す個々のポジションに対して、このまさしくそのクラスタIDのための別個の副マトリクスが、マトリクス42の中の個々のポジションに、信号で伝えられて置かれる。従って、`max_layer_id_ext`は、`vps_max_layers_minus1`クラスタのそれぞれの内部の層の数に一致する。

【0054】

図8の場合、第1相互依存構文構造24は、図7の例と同じ方法で送られる：個々のクラスタIDに対して、クラスタID当たり1つのフラグは、個々のクラスタIDと等しい又はより小さい。フラグ`cluster_dependency_sets_present_flag`46は、上で既に概説していた構文要素`general_direct_ext_dependency_flag`を使って、図7の第1オプション間を切り換える。第2相互依存構文構造26の実例の数が、構文要素`direct_ext_dependency_flag`を使って、データストリームの中に提出されることに従う異なるモードが、`number_of_cluster_dependency_sets`と称される構文要素を使って、明示的に信号で伝えられる。これらの実例は、その時、個々の実例が構文要素`included_nuh_layer_id`50を使って適用されるクラスタIDを明示的に記載することによって、これらの実例が適用されるクラスタIDに関連する。この手段によって、第2相互依存構文構造26の実例34は、個々のクラスタIDに対して提出されない。むしろ、第2相互依存構文構造26の実例のより小さい数のみが、1と等しい`cluster_dependency_sets_present_flag`の場合に提出される。個々の実例は、実例に関連するクラスタIDに関連している。また、異なるクラスタの相互依存するペアが関係する限り、欠陥規則がマトリクス42をそのようなポジション（クラスタ対角線の下側）で満たすために適用される。

【0055】

図9の例に従って、第2相互依存構文構造の信号は、第1相互依存構文構造について差し挟まれた方法で、例示的に信号で伝えられる。しかし、この差し挟みは代替の実施の形態に従って除外される。ともかく、図9の実施の形態において、副マトリクス38または40（すなわち、第2相互依存構文構造26の実例34）は、第1相互依存構文構造24が相互クラスタ依存を標示する（等しい又は等しくない）クラスタIDの個々のペアに対して提出される。構文例から引き出されるように、副マトリクス40の送信は、副マト

10

20

30

40

50

リクス38より多いビット`direct_ext_dependency_flag`を消費する。これは条件条項52と条件条項54からそれぞれ引き出すことができる。すなわち、クラスタ内部の相互依存規則のために第2相互依存構文構造26の実例を送信することにおいて、カウンター*i*によって示されたリファレンスしているクラスタIDは、カウンター*j*によって示されたリファレンスされたクラスタIDに等しく、`direct_ext_dependency_flag`は、エンハンスメント層IDの結合のために、リファレンス及びリファレンスしている層（リファレンスされた層（ここでは*l*）のエンハンスメント層IDが、リファレンスされた層（ここでは*k*）のエンハンスメント層IDのエンハンスメント層IDより小さい）に対してのみ送信される。すなわち、副マトリクス38の`direct_ext_dependency_flag`は、副マトリクス38の対角線の下側のポジションに対してのみ送信される。別の場合において、すなわち、相互依存するクラスタペア*i j*のための第2相互依存構文構造26の実例に対して、完全な副マトリクスが送信される。すなわち、フラグ`direct_ext_dependency_flag`が、副マトリクス40のそれぞれのポジション毎に送信される。

【0056】

図10の実施の形態は、図8及び図9の実施の形態の特定の特性：完全な副マトリクスを結合する。すなわち、第2相互依存構文構造26の実例が、つまり数の中の`number_of_cluster_dependency_sets`が提出される。そして、その時、これらの実例のそれぞれは、マトリクス36のグリッドサイトに関連する。ここで、例えば、第2相互依存構文構造26の個々の実例が、第1相互依存構文構造24がそれによってクラスタ独立性の存在を示すために（符号56と比較してください）それらの間で適用される。

【0057】

言い換えると、図9について説明したように、第2相互依存構文構造26の複数の実例34は、多層データストリームから、つまり、ベース層IDによって表現できる数値である*N*を持つ値 $0 < k - j - n < N$ のペア（*j*、*k*）に対して読み取られる。*n*は*N*に等しいように設定される。しかし、また、ここに説明されるように、実際に使われたクラスタ濃度の明示的な信号は、送信オーバーヘッドを制限するために使われる。ペア（*j*、*k*）は詳察されて（図9の符号24で*i*と*j*に亘るループを見てください）、そして、第2相互依存構文構造を読み取ることが、このペア間の相互依存の存在または不存在を示している第1相互依存構文構造に条件付きで依存しているペア（*j*、*k*）に対して実行される又は抑制される（図9の「`if (direct_dependency_flag[i][j])`」を見てください、ここで、*i*、*j*は*j*と*k*にそれぞれ一致する）。*j* = *k*を持つペア（*j*、*k*）に対して読み取る第2相互依存構文構造26の実例34は、ベース層ID *j*を持つそれらの間の層が、ベース層ID *j*を持つそれらの間の別の層に依存する相互層予測であることを示す。*j* > *k*を持つ異なる値のペア（*j*、*k*）に対して読み取る第2相互依存構文構造26の実例34は、ベース層ID *j*を持つそれらの間の層が、ベース層ID *k*を持つそれらの間の層に依存する相互層予測であることを示す。しかし、図7と同様に、第2相互依存構文構造26の1つの実例が、多層データストリームから読み取られることもある。第2相互依存構文構造26の1つの実例は、個々のクラスタ値ペア（*j*、*j*）に対して、共通して示す。ベース層ID *j*を持つそれらの間の層は、ベース層ID *j*を持つそれらの間の別の層に依存する相互層予測である。図7の場合、仮に、`unique_cluster_dependencies_flag`が0に等しい、または、*j - k*を持つペア（*j*、*k*）に対して共通しているならば、第1相互依存構文構造24は、個々のペア間の相互依存の存在を示す。ベース層ID *j*を持つそれらの間の層は、ベース層ID *k*を持つそれらの間の層に依存する相互層予測である。拡張層ID22によって表現できる数値を*M*と示し、第2相互依存構文構造は、ベース層ID *i*を持つそれらの間の層が、ベース層ID *i*を持つそれらの間の別の層に依存する相互層予測であることを示すために、拡張値 $0 < q < p - m < M$ の個々のペア（*p*、*q*）に対して1つの2進数の値を含み、そして、拡張値*p*およびベース層ID *j*を持つ層が、拡張値*q*

10

20

30

40

50

およびベース層ID k を持つ層に依存する相互層予測であることを示すために、拡張値 $0 < p, q \leq m < M$ の個々のペアに対して1つの2進数の値を含む。また、明細書の中の別の場所で説明されるように、 m が M と等しいように設定される。しかし、また、ここに説明されるように、実際に使われた拡張濃度の明示的な信号は、送信オーバーヘッドを制限するために使われる。さらに、図8と図10に示すように、インデックス構文構造50が、第2相互依存構文構造の実例をベース層ID値の個々のペア(j, k)に関連付けるために使用される。ここで、図10は、関連構文構造がペア(j, k)毎にフラグを含み、関連構文構造の構造解析が、第1相互依存構文構造がペア(j, k)間の相互依存の不存在(独立性)を示す値 $0 < k \leq j \leq n < N$ のスキップペア(j, k)を含むことを明らかにする。

10

【0058】

すなわち、上のセクションの中で説明されたものは、「層識別子クラスタ化とクラスタ依存信号化」として示される。これに関して、装置はビデオデコーダである。このように動作するビデオデコーダは、復号化に対して、(拡張された)層IDによって識別されたビットストリームのパケットを選択することができる。しかし、装置は、代わりに、ネットワーク利用(デコーダなどについての知識)のような外部のファクターに基づいた(拡張された)層IDによって識別されたビットストリームのパケットを捨てることのできるネットワーク要素であってもよい。

【0059】

当然、上で概説された概念は、概念によって改善されるべき符号器の既存のエンコーダも詰め込む。図2の1つのようなエンコーダは、情報量の異なるレベルと一致している異なる層で、相互層予測を使って、ビデオ素材を多層データストリーム10の中に符号化するように構成される。多層データストリームは、それぞれが異なる層のうちの1つと関連する複数のパケット12を含む。各層は、ベース層ID20、または、ベース層ID20及び拡張層ID22によって索引を付けられ、多層データストリームの中に挿入される。第1相互依存構文構造24は、ベース層ID20によって表現できる異なる値のペア間の相互依存を示す。そして、第2相互依存構文構造26は、拡張層ID22によって表現できる異なる値のペア間の相互依存を示す。第1および第2相互依存構文構造をそのように設定することで、第1および第2相互依存構文構造に基づいて、異なる層間の可能な相互層予測依存を明らかにする相互層依存記述14が、組み立て可能である。

20

30

【0060】

次に記述される詳細は、効率的な方法で、層関連の情報を信号で伝える可能性に関する。そのような層関連の情報は、例えば、図2から図10について概説した相互層依存に関する情報を含む。出発点として、しばしば上に示された状況が、個々のパケット206によって構成された層識別構文要素構造208(図2と比較してください)が、図11において描かれたベース層IDフィールド108と拡張層IDフィールド112とから成ることによって従って、使われる。図11は、図2の中に示されるネットワーク要素218または多層ビデオデコーダ220などの多層データストリームを処理するための装置100を示す。装置100は、一連のパケット(そのそれぞれが層識別構文要素を含む)から成る多層ビデオ信号を受信するように構成されたレシーバー102を含むとして示される。

40

【0061】

レシーバー102によって受信された多層ビデオ信号104(図2の符号204と一致している)は、一連のパケット106(図2の符号206と一致している)を含む。各パケット106は、少なくともベース層IDフィールド108を含む層識別構文要素構造(図2の符号208と一致している)を含み、ハイレベル構文要素またはベース層IDフィールド108または拡張層IDフィールド112の値に無条件に又は条件付きで依存する。

【0062】

例えば、パケット106は、多層ビデオ信号104が副分割されるNALユニット、スライス、タイル、画像、または別のパケットである。また、装置100に含まれた層識別

50

エクステンダー 110 は、部分 113 のような複数のパケット 106 を含む多層ビデオ信号の予め決められた部分に対して、多層ビデオ信号の予め決められた部分 113 を含むパケット 108 の拡張層 ID フィールド 112 の最大限に仮定された値を示す多層ビデオ信号から最大構文要素を読み取る。予め決められた部分 113 は、例えば、符号化されたビデオシーケンス、大きな塊、画像のグループなどである。最大構文要素 114 は、例えば VPS NAL ユニットのよう部分 113 の特定のパケット内に含まれる。予め決められた部分 113 内のパケットのそれぞれに対して、エクステンダー 110 は、符号 108、そして、条件付きで符号 112 から成る層 ID 構文構造に基づいて個々のパケットのために層 ID を決定する。例えば、構文要素 108 と 112 の両方の値の連結は、層 ID を生じる。

10

【0063】

エクステンダー 110 は、上で説明された最大構文要素を使って、拡張層 ID フィールドのビット長さを引き出す。代わりに、特別な構文要素が、この終わりに使用されてもよい。代替でも、拡張層 ID フィールド 112 を表すためのビット数は、欠陥（デフォルト）によって固定される。

【0064】

エクステンダー 110 は、その時、最大限に仮定された値に基づいて、部分 112 内の層の最大数を決定する。例えば、エクステンダー 110 は、部分 113 に対して、部分 113 内のパケット 106 のベース層 ID フィールド 108 のための最大限に仮定された値を示すデータストリーム 104 から構文要素も読み取り、そして、層の最大数を決定するために両方の最大限に仮定された値を結合する。

20

【0065】

エクステンダー 110 は、層の最大数のそれぞれに対して、多層ビデオ信号の層特徴付け構文部分を、層の最大数と等しい回数繰り返して構文解析することによって、最低 1 つの特徴を決定する。利点的に、データストリーム 104 は、それぞれの可能な値に対して（すなわち、拡張層 ID フィールド 112 の全体の濃度全体に対してではなく、この濃度の外の実際に使われた副部分に対して）、最低 1 つの特徴を信号で伝えることを必要としない。さらに、最大構文要素 114 は、丁度記載されたように、データストリーム 104 内の拡張層 ID フィールド 112 の表現ビット数を引き出すように使われさえする。例えば、「特徴」は、別の層または同類に対する相互層予測である。

30

【0066】

すなわち図 11 について、データストリーム 204 / 104 内で、重要性の予め決められた部分 113 内の全てのパケット 106 / 206 を考慮するとき、拡張層 ID フィールドの表現できる状態の外にいくつが実際に仮定されるかの明示的な信号化を送ることが可能であることが説明される。例えば、拡張層 ID フィールド 112 が N ビットのフィールドであるとする、 2^N 個の拡張層 ID 120 が拡張層 ID フィールド 112 によって表現できる。フィールド 112 の長さは、固定されるか、または信号で伝えられて、データストリーム内で、図 12 の点線 122 を使って示された所定のハイレベル構文要素の方法によって設定される。しかし、いくつかの理由のために、利用できる拡張層 ID の全てが、重要性の予め決められた部分 113 内で使われない。例えば、図 12 において、7 個の表現できる拡張層 ID のみが使われることが描かれる、すなわちハッチングの方法によって描かれる。

40

【0067】

図 11 に従って、最大構文要素 114 が、拡張層 ID フィールド 112 の実際に使われた / 仮定された値のこの数を示す。この構文要素 114 のため、拡張層 ID フィールド 112 の表現できる値の完全な数に対してというよりも、むしろ拡張層 ID の実際に仮定された数についてのみ層特定情報をデータストリームの中に挿入することが実現できる。従って、図 12 の符号 122 で説明されるように、多層ビデオ信号を構文解析している装置は、例えば、拡張層 ID フィールドの実際に利用できる / 表現できる値の完全な数よりもむしろ構文要素 114 によって示された層の最大数に等しい回数だけ、多層ビデオ信号の

50

所定の層特徴付け構文部分 1 2 4 を構文解析する。データストリームによって消費されたサイド情報量は、従ってより低く保持される。例えば図 6 から図 1 0 の例において、拡張層 ID フィールド 1 1 2 の表現できる値の最大数と等しいようにこの値 (max_layer_id_ext) を設定するよりもむしろ、構文要素 1 1 4 によって示される拡張状態の最大限に仮定された数と等しいように max_layer_id_ext を設定することが実現できる。従って、「特徴付け構文部分」1 2 4 は、以前に提出されたフラグ $\text{direct_ext_dependency_flag}$ によって具体的に示される。

【0068】

上で既に概説したように、フィールド 1 1 2 のビット長さの存在 / 信号化はオプションである。ともかく、構文要素 1 1 4 が実際に使用した / アクティブな拡張層 ID の数を示す粒状性 / 忠実性は、構文要素 1 2 2 によって設定された拡張層 ID フィールド 1 1 2 の利用できる / 表現できる値の数をより粗く設定することに比べて増大する。好ましくは、構文要素 1 1 4 が拡張層 ID フィールド 1 1 2 の実際に仮定された値の数を示す忠実性または粒状性は、実際に仮定された値の正確な数を示すために極めて十分である。代わりに、ものの間のどこかの忠実性 / 粒状性、および、構文要素 1 2 2 によって提供されたフィールド 1 1 2 の利用できる / 表現できる値の最大数の信号化の忠実性 / 粒状性が同様に実現できる。すなわち、最大構文要素 1 1 4 は、拡張層 ID フィールドのビット長さである n を持つ $(n - 1)^2$ より小さい又は等しいユニットの中の拡張層 ID フィールド 1 0 8 の最大限に仮定された値を示す。

【0069】

従って、図 1 1 と図 1 2 に従って、ネットワーク要素または多層ビデオデコーダなどの装置が、レシーバー 1 0 2 およびエクステンダー 1 1 0 を含むことが説明される。レシーバー 1 0 2 は、一連のパケット 1 0 6 から成る多層ビデオ信号 1 0 4 を受信するように構成されている。パケット 1 0 6 のそれぞれは、ベース層 ID フィールドおよびことによると条件付きのみで拡張層 ID フィールド 1 1 2 から成る層識別構文要素構造を含む。層識別エクステンダー 1 1 0 は、一連のパケット 1 0 6 の外の複数のパケットを含む多層ビデオ信号の予め決められた部分 1 1 3 に対して、多層ビデオ信号 1 0 4 から最大構文要素 1 1 4 読み取るように構成されている。最大構文要素 1 1 4 は、予め決められた部分 1 1 3 によって構成されたパケット 1 0 6 の間の拡張層 ID フィールド 1 1 2 の最大限に仮定された値を示す。エクステンダー 1 1 0 は、予め決められた部分 1 1 3 内のパケット 1 0 6 のそれぞれに対して、上で概説されたフィールド 1 0 8 と 1 1 2 のベース層 ID と拡張層 ID とを連結することなどによって、層識別構文要素構造に基づいた個々のパケット 1 0 6 のための層 ID を決定する。エクステンダー 1 1 0 は、また、最大限に仮定された値に基づいた予め決められた部分 1 1 3 内の層の最大数を決定し、層の最大数のそれぞれに対して、層特徴付け構文部分 1 0 4 を、層の最大数と等しい回数繰り返して構文解析することによって、最低 1 つの特徴を決定する。「層の最大数」は、図 1 1 と 1 2 の実施の形態を図 2 ~ 図 1 0 について上記で概説した実施の形態と結合するとき、クラスタ当たり層の最大数を示す。この場合、「層の最大数」の決定は、例えば、最大限に仮定された値を「層の最大数」として直接に採用する。「最低 1 つの特徴」は、第 2 相互依存構文構造内の相互層依存を示しているフラグの数である。しかし、特徴の別の例は、個々の層 ID に対して設定された一致している符号化パラメータのデータストリーム内の信号化でもある。その場合、「層の最大数」は、使われた又は表現できるベース層 ID の数 \times 最大限に仮定された値、すなわち実際に使用された / 仮定された拡張層 ID の数と等しいように決定される。別の例は同様に実現可能である。ともかく、構文要素 1 1 4 の方法によって実際に仮定された値の伝達は、貴重なサイド情報ビットレートを保存することを可能にする。

【0070】

図 1 1 と図 1 2 の概念は、上で既に概説されたように、図 2 ~ 図 1 0 について上に概説した概念と結合される、または、以前に説明された概念を使わないで単独で使われる。

【0071】

しかし、図 2 ~ 図 1 0 について前に提出された上記の説明と同様に、以下の説明は、H

10

20

30

40

50

ＥＶＣの拡張などの既存のビデオ符号器の中に、図１１と図１２の概念をいかにして築くかの可能性を明らかにする。

【００７２】

ＨＥＶＣにおいて、層識別子変数（`LayerId`と称する）は、ビデオおよびメタデータパケットのヘッダー、例えばビデオまたはメタデータパケットの大きな塊のそれぞれの中において信号で伝えられた`nuh_layer_id`および追加の情報の明示的に信号で伝えられた層識別子から引き出され、その識別のために使われるように、特定のデータパケットの新しい可変の`LayerId`を形成する。

【００７３】

図１１と図１２に従って信号で伝えられた追加の情報は、特定のビデオまたはメタデータパケットがエンハンスメント層（向上層）に属し、それ故、レガシー（古くて機能的には劣るけれども普及していて替えることができない）装置のみベース層と互換性を維持することを与えられた追加の構文要素（ビデオまたはメタデータパケットのヘッダーまたはペイロードのいくつかの部分内の`layer_id_ext`と比較してください）を含む。ビデオまたはメタデータパケットが、ベース層に属すけれども、有能な装置を処理しているエンハンスメント層によって処理されるとき、デフォルト値は`layer_id_ext`に対して仮定される。

【００７４】

図１３は、ＨＥＶＣ符号化ビデオのエンハンスメント層（`nuh_layer_id > 0`と比較してください）のビデオデータパケットの中に含まれるスライスヘッダーの中の構文要素`layer_id_ext`として、追加の情報を信号で伝えるスライスセグメントヘッダーのための例を与える。

【００７５】

数学的な操作は、`nuh_layer_id`と`layer_id_ext`を、より大きい識別子空間を許し、かつ`nuh_layer_id`と`layer_id_ext`の２つの値の特別な組み合わせに対して特別である`LayerId`の中に結合するために使われる。数学的な操作は、例えば以下のように、`LayerId`変数の`MSB`として`nuh_layer_id`のビットが、そして、`LayerId`変数の`LSB`として`layer_id_ext`のビットが使われる（または、逆もまた同様である）。

【００７６】

可変の`LayerId`は次の通り決定される。そこで、`LengthOfExtension`はビットについての`layer_id_ext`構文要素の長さである。

【００７７】

$$LayerId = (nuh_layer_id \ll LengthOfExtension) + layer_id_ext$$

【００７８】

図１１と図１２の概念は、送られたデータ量を効率的に利用するために、ビデオまたはメタデータパケットのヘッダーまたはペイロードのいくつかの部分の追加の情報（上の`LengthOfExtension`と比較してください）を伝えるために使用されるビットの量の明示的な信号化を含む。図１４は、ＨＥＶＣビデオパラメータセット拡張構文などのメタデータパケットにおいて信号で伝えている`LengthOfExtension`値の例示的な構文と意味を与える。

【００７９】

`layer_id_ext_len`は、`LayerId`範囲を拡張するために使われるビット数を示す。

【００８０】

図１１と図１２の概念は、符号化されたビデオビットストリーム（例えば`max_layer_id_ext`）において使われた`layer_id_ext`構文要素の最大値の明示的な信号化を含む。それに基づいて、ビデオまたはメタデータパケットのヘッダーまたはペイロードのいくつかの部分の追加の情報（上の`LengthOfExtension`

10

20

30

40

50

nと比較してください)を伝えるために必要なビットの最低可能量が、送られたデータ量を効率的に利用するために引き出される。図15は、HEVCビデオパラメータセット拡張構文などのメタデータパケットにおいて信号で伝えているlayer_id_ext構文要素値の最大値の構文と意味を例示的に与える。

【0081】

max_layer_id_extは、符号化されたビデオシーケンス内のスライスヘッダーの中のlayer_id_ext構文要素の最大値を示す。

発明の上記の拡張に依存しているlayer_id_ext構文要素の意味の2つの例示的なバリエーションが、以下において与えられる。

layer_id_extは、層を識別するために、nuh_layer_idとの結合において使用される。構文要素layer_id_extは、layer_id_ext_lenビットを使って符号化される。存在しないとき、その値は0と推定される。

layer_id_extは、層を識別するために、nuh_layer_idとの結合において使用される。構文要素layer_id_extは、符号化されたCeil($\log_2(\max_layer_id_ext)$)ビットである。存在しないとき、その値は0と推定される。

【0082】

HEVCビデオ符号器規格のパラメータセットまたはSEIメッセージの層識別子は、図16a~図16cにおいて与えられた依存を構造解析すること無く拡張される。

【0083】

層にされた符号化シナリオにおいて、図17のビデオパラメータセット拡張のための構文テーブルで例示的に与えられるように、層識別子またはその最大数に関連する信号化は、正規の明示的な層識別子空間(nuh_layer_idの値範囲と比較してください)の代わりに拡張層識別子空間(LayerIdの値範囲と比較してください)をカバーするために適合される。ここで、可変のVpsMaxNumLayersが、次の通り例示的に与えられる。

拡張層の範囲の中の層の最大数を規定している可変のVpsMaxNumLayersは、 $(vps_max_layers_minus\ 1) * 2^{layer_id_ext_len} + 1$ と等しく設定される。または

拡張層の範囲の中の層の最大数を規定している可変のVpsMaxNumLayersは、 $(vps_max_layers_minus\ 1) * max_layer_id_ext + 1$ と等しく設定される。

【0084】

layer_id_in_nalu[i]は、i番目の層のVCL NALユニットと関連したLayerId値の値を規定する。0からVpsMaxNumLayers-1の範囲内のiに対して、包括的に、存在しないとき、layer_id_in_nalu[i]の値はiに等しいと推定される。

【0085】

iが0より大きいとき、layer_id_in_nalu[i]は、layer_id_in_nalu[i-1]より大きい。0からvps_max_layers_minus 1の範囲内のiに対して、包括的に、可変のLayerIdInVps[layer_id_in_nalu[i]]はiと等しく設定される。

【0086】

dimension_id[i][j]は、i番目の層のj番目の現在のスケーラビリティ次元タイプの識別子を規定する。dimension_id[i][j]の表現のために使われたビット数は、dimension_id_len_minus 1[j] + 1ビットである。dimension_id[i][j]が、0からNumScalabilityTypes-1の範囲内のjに対して、存在しないとき、包括的に、dimension_id[i][j]は、 $((layer_id_in_nal_u[i] \& ((1 \ll dimBitOffset[j+1]) - 1))) \gg dimBitOffset[j]$

10

20

30

40

50

」)に等しいと推定される。

【0087】

i 番目の層の `smIdx` 番目のスケーラビリティ次元タイプの識別子を規定している可変の `ScalabilityId[i][smIdx]`、i 番目の層のビュー識別子を規定している可変の `ViewId[layer__id__in__nalu[i]]`、および、i 番目の層がビュースケーラビリティ拡張層であるかどうかを規定している可変の `ViewScaleExtLayerFlag` は、次の通り引き出される。

```
for ( i = 1 ; i <= VpsMaxNumLayers ; i ++ ) {
    lId = layer__id__in__nalu[i]
    for ( smIdx = 0 , j = 0 ; smIdx < 16 ; smIdx ++ )
        if ( i != 0 ) && scalability__mask[smIdx] )
            ScalabilityId[i][smIdx] = dimension__id
[ i ][ j ++ ]
    else
        ScalabilityId[i][smIdx] = 0
    ViewId[lId] = ScalabilityId[i][0]
    ViewScaleExtLayerFlag[lId] = ( ViewId[lId]
!= ViewId[0] )
}
```

10

【0088】

20

当然、上で概説された概念は、概念によって改善される符号器の既存のエンコードも詰め込む。図2のようなエンコードは、一連のパケット106から成る多層ビデオ信号104の中にビデオ信号を符号化するように構成されている。パケット106のそれぞれは、ベース層IDフィールド108と拡張層IDフィールド112から成る層識別構文構造を含む。さらに、このエンコードは、複数のパケットを含む多層ビデオ信号の予め決められた部分113に対して、最大構文要素114を多層ビデオ信号の中に挿入し、多層ビデオ信号の予め決められた部分113を含むパケットの拡張層IDフィールド108の最大限に仮定された値を示し、最大限に仮定された値に基づいて多層ビデオ信号の予め決められた部分内の層の最大数を決定し、層の最大数のそれぞれに対して、多層ビデオ信号の層特徴付け構文部分124を層の最大数と等しい回数繰り返し書くことによって、多層ビデオ信号104の中に最低1つの特徴を信号で伝えるように構成されている。例えば、エンコードは、部分113の符号化を終えるまで、ストリーム104の出力を延期する。

30

【0089】

以下の図面は、多層データストリームに含まれたパケットの層IDを信号で伝える概念、および、どのような推論のためでも、0などのベース層IDの特定の値が、そこにおいて拡張層IDを付加した可能性から除外されることを要求することが必要な場合において、バックワード互換性を解決すること無く拡張メカニズムを使って、この層IDを効率的に信号で伝えることを達成する可能性を扱う。後者の必要性は以下の非対称を生じる：ベース層IDの殆んど全ての可能な値に対して、拡張層IDを付加することが実現できる。これは以下の非対称を生じる：ベース層IDの殆んど全ての値に対して、拡張層IDの付加が実現でき、従って、ベース層IDの殆んど全ての値の数×層ID値を表現可能な拡張層IDの表現可能な値の数を生じる。しかし、限定されたベース層ID値は、どのような拡張層IDでも付加する可能性を除く。従って、この非対称が受け入れられるか、または、ベース層IDの禁じられた値が、拡張メカニズムを使う場合において使われないかのいずれかである。以下に発表された解決は、より効率的な方法でこの問題を克服する。

40

【0090】

今回、この次の概念の説明は、ちょうど概説された制限が正確に適用されるHEVC規格の可能な拡張について直ちに概説される。しかし、以下で概説された概念は、例えば層識別拡張メカニズムの上に同様な制限を課す別の符号器にも移送可能である。

【0091】

50

HEVCの制限は、上で概説したように、ベース層が拡張できないことである。ゆえに、HEVC規格に従うために、層id拡張値`layer_id_ext`は、0と等しい`nuh_layer_id`を持つ層に対して、信号で伝えることができない。従って、0と等しい`nuh_layer_id`に関連したクラスタは、ベース層自身のみを含むことができる。そのような異なるクラスタの中の層の等しくない数は不適当である。

【0092】

問題は、以下に概説された概念によって解決できる。それは、クラスタおよびクラスタ内の層id拡張値を、上の説明とは異なる方法で引き出す。それは、0と等しい`nuh_layer_id`に対して複数の層を持つことができないので、複数の層を持たない制限は、0と等しくない別の`nuh_layer_id`値に拡張される。これらの限定された`nuh_layer_id`値に属する層は、そのとき、層の等しい数を持つクラスタを形成するために、無制限の`nuh_layer_id`値からの層と結合される。

10

【0093】

基本的なアイデアは、`nuh_layer_id`が、例えばビットマスキングによって2つの値に分離されることである。1つの値は、クラスタを決定するために用いられる。この方法の異なる`nuh_layer_id`値は同じクラスタに属することができる。もう1つの値は、`layer_id_ext`値の存在を決定し（それゆえ、仮に、複数の層が`nuh_layer_id`値に属することができるならば、決定し）、そして、`LayerId`派生において`layer_id_ext_value`に追加のオフセットを決定するために利用される。

20

【0094】

基本的なアイデアの実施の形態のための例は、以下において与えられ、図18において説明される。従って、図18は、3と等しい`max_layer_id_ext`（2と等しい`LengthOfExtension`）の例示的な場合に対して、`layer_id_ext`と`nuh_layer_id`と`LayerId`との間の関係のための例を提供する。

【0095】

この例のみににおいて、6ビット`nuh_layer_id`値のうち5個の最も重要なビットが、クラスタを決定するために用いられる。ゆえに、異なる最も重要でないビットを持つ2つの`nuh_layer_id`値が、1つのクラスタに属している。

30

【0096】

仮に、最も重要でないビットが0と等しいならば、`layer_id_ext`値は、`nuh_layer_id`および`uh_layer_id`に属する1つの層のみにに対して、信号で伝えられない。定義によって、この層は、5個の最も重要なビットによって規定されたクラスタ内の0と等しい`layer_id_ext`を持つ層である。さもなければ、仮に、最も重要でないビットが1と等しいならば、複数の層は`nuh_layer_id`値に属する。個々の層に対して、`layer_id_ext`値は明示的に信号で伝えられ、個々の層は5個の最も重要なビットによって規定されたクラスタに属する。

【0097】

図19の構文テーブルは実施の形態のために使われる。それはパケット206のヘッダー、すなわちスライスセグメントのヘッダー、または、より精密なその破片を示す。

40

【0098】

`nuh_layer_id`の最も重要でないビットが0と等しいとき、`Layer_id_ext`は存在しない。それが存在しないとき、それは0に等しいと推定される。可変の`LayerId`は次の通り決定される。ここで、`LengthOfExtension`は、ビットについての`layer_id_ext`構文要素の長さである。

$$\text{LayerId} = ((\text{nuh_layer_id} > 1) << \text{LengthOfExtension}) + \text{layer_id_ext}$$

【0099】

別の実施の形態において、可変の`add_layer_id_ext_len`は、ビット

50

トストリ - ム、例えば V P S において信号で伝えられる。(前に説明した実施の形態が、1 と等しい `add_layer_id_ext_len` を持つこの実施の形態の特別な場合であることに注意してください)。6 ビット `nuh_layer_id` 構文要素の最も重要なビット (6 - `add_layer_id_ext_len`) は、クラスタを決定するために使われる。最も重要でないビット `add_layer_id_ext_len` は、`layer_id_ext` が推定されるかどうかを規定する。さらに、それらは `layer_id_ext` 値に追加のオフセットを規定する。図 20 a および図 20 b の構文テーブルは、実施の形態を示すために用いられる。

【0100】

実施の形態において、`nuh_layer_id` の最も重要でないビット (`add_layer_id_ext_len`) が 1 と等しいとき、`layer_id_ext` のみが存在する。`layer_id_ext` が存在しないとき、それは 0 に等しいと推定される。可変の `LayerId` は次の通り決定される。ここで、`LengthOfExtension` は、ビットについての `layer_id_ext` 構文要素の長さである。

`LayerId =`
 $((\text{nuh_layer_id} > \text{add_layer_id_ext_len}) < \text{LengthOfExtension}) + \text{layer_id_ext} + \text{layerIdExtOffset}$

【0101】

従って、図 18 ~ 図 20 b について上で概説された概念に従って、図 2 において描かれた多層ビデオデコーダのネットワーク要素などの装置は、相互層予測を使って、ビデオ素材が、情報量の異なるレベルに一致している異なる層で符号化される多層データストリームを処理するように構成される。装置は、多層データストリームを含むパケットのそれぞれに対して、以下のことを行う。個々のパケットは、上で既に説明したように、異なる層のうちの 1 つと関連している。

【0102】

特に、図 21 に示すように、装置は、多層データストリームから、または、より精密であるように現在のパケットから、ベース層 ID フィールド 308 を読み取る。ベース層 ID フィールドは第 1 副フィールド 308 a と第 2 副フィールド 308 b とを含む。例えば、第 1 副フィールドは、フィールド 308 の 1 つ以上の最も重要でないビットによって形成される。一方、第 2 副フィールド 308 b は、フィールド 308 の別の最も重要なビットによって形成される。図 20 b の例において示されるように、`add_layer_id_ext_len` と称される高レベル構文要素 310 は、副フィールド 308 a の長さを定義する。

【0103】

ステップ 312 において、装置はベース層 ID フィールド 308 を読み取った後、ステップ 314 において、副フィールド 308 a が、予め決められた基準を満たすかどうかをチェックする。例えば、仮に、副フィールドが 1 つであるか、または「全く同じシーケンス」であるならば、基準は満たされる。予め決められた基準が満たされる場合、装置は、ステップ 318 において、データストリームから、または、より精密であるように、現在のパケットから拡張層 ID フィールド 316 を読み取る。そして、ステップ 320 において、拡張値が拡張値のドメインの第 1 副セット内にあるように、拡張層 ID フィールドを使って拡張値を引き出す。ステップ 320 における引き出しは、フィールド 316 内のビット表現の単なるコピーと一致する。拡張層 ID フィールド 316 内の値が 0 などの特定の値を仮定しないことは、データストリームの要件である。さらに以下で概説されるように、ベース層 ID フィールド 308 の副フィールド 308 a が基準を満たさない場合、後者の値が確保される。

【0104】

ステップ 320 の後、装置は、第 2 副フィールド 308 b の可能な値のドメインからクラスタ値のドメイン上にマッピングすることによって、クラスタ値を第 2 副フィールド 3

10

20

30

40

50

0 8 b から引き出す。現在のパケットが関連する層、すなわち、フィールド 3 0 8 および 3 1 6 を構成している層が、ステップ 3 2 4 において、ステップ 3 2 0 と 3 2 2 において引き出されたクラスタと拡張値を使って索引を付けられる。例えば、クラスタおよび拡張値は、例えば最も重要な数字のためのクラスタ値を使って、お互いに単に連結される。副フィールド 3 0 8 b に基づいたクラスタ値 3 2 2 の引き出しは、また、例えば、クラスタ値の 2 進数の表現として、副フィールド 3 0 8 b のビットの直接的な使用から成る。

【 0 1 0 5 】

しかし、仮に、ステップ 3 1 4 のチェックが、予め決められた基準が合致しないならば、装置は、現在のパケットに対して多層データストリームから拡張層 ID フィールドを読み取ることを控える。むしろ、装置は、ステップ 3 2 6 において、拡張値を 0 に設定することなどによって、拡張値のドメインの第 1 副セットに解体する値の上に拡張値を設定し、そして、クラスタ値のドメインの上に第 2 副フィールド 3 0 8 b の可能な値のドメインからマッピングすることによって、クラスタ値を第 2 副フィールド 3 0 8 b から引き出す。ステップ 3 2 2 と 3 2 8 の引き出しは、同じマッピングを使って実行される。次に、現在のパケットの層 ID は、ステップ 3 2 6 と 3 2 8 において設定されて引き出されたクラスタと拡張値から決定される。ここで、仮に、クラスタと拡張値がステップ 3 2 0 と 3 2 2 において引き出されるならば、層 ID の決定が同じ方法で再び実行される。

【 0 1 0 6 】

図 2 1 の説明を図 1 8 の実施の形態に関連させて、第 1 副フィールド 3 0 8 a は、1 ビットの長さであり、特にフィールド 3 0 8 の LSB である。仮に、この LSB 3 0 8 a が勝ち取られて、拡張層 ID フィールド 3 1 6 にわたる拡張値の前記ドメインが { 0、1、2、3 } であり、図 1 8 の場合において、layer_id_ext、すなわちフィールド 3 1 6 が 2 ビットの長さであるならば、予め決められた基準が満たされる。従って、拡張値のドメイン、すなわちステップ 3 2 0 における引き出しの共通ドメインの第 1 副セットは、{ 1、2、3 } である。特に、ステップ 3 2 0 における引き出しは、図 1 8 の例の中で、拡張層 ID フィールドに適用されたユニットのマトリクスの方法によって実施される。代わりに、いくつかの別の計算が、第 1 副フィールド 3 0 8 a の値をユニットのマトリクス応用の結果に追加するように使われる。従って、ステップ 3 2 6 において、拡張値は、{ 1、2、3 } に解体する値、すなわち 0 に設定される。引き出し 3 2 2 と 3 2 8 において使われたマッピングは同じであるけれども、代わりに異なってもよい。マッピングは特にバイジェクティブ (bijective) である。

【 0 1 0 7 】

ステップ 3 2 4 の中のインデックスの決定は、図 1 8 において説明されるように、より重要な数字を設定するためのクラスタ値と、より重要でない数字を設定するための拡張値とを使用する。従って、図 1 8 において説明したように、layer_id_ext フィールド、すなわち拡張層 ID フィールド 3 1 6 は、nuh_layer_id、すなわちベース層 ID フィールド 3 0 8 の LSB が 1 である場合に単に存在する。その他の場合は、layer_id_ext フィールドは 0 であると推定される。従って、layer_id_ext と nuh_layer_id の結合に対して層 ID のマッピングは、完全に対称であり、未使用のビット表現は存在しない。layer_id_ext の推定された値は、layer_id_ext を移送することによって、データストリームの中で利用される。その結果、layer_id_ext の移送のためのデータストリームの中に亘るビットが、実のところ、layer_id_ext が 3 つの異なる値 { 1、2、3 } を単に仮定するという事実を説明するように減少する。例えば、2 進数の単項符号は、エントロピー符号化 / 復号化を使って、データストリームの中に符号化され、かつ、そこから復号化された 2 進数のストリングスの上に、{ 1、2、3 } の 3 つの要素をマップするために使用される。

【 0 1 0 8 】

従って、相互エイリアスが後ろの概念について記述される。相互層予測を使って、情報の異なるレベルに一致している異なる層でビデオ素材が符号化される多層データストリ

10

20

30

40

50

ームを処理するように構成された装置。それぞれが異なる層のうちの1つと関連する複数のパケットを含む多層データストリーム。多層データストリームのパケットのそれぞれに対して、多層データストリームからベース層IDフィールドを読み取るように構成された装置。第1副フィールドおよび第2副フィールドを含むベース層IDフィールド。ベース層IDフィールドの第1副フィールドが、予め決められた基準を満たすかどうかについてのチェック。

【0109】

仮に、ベース層IDフィールドの第1副フィールドが、予め決められた基準を満たさなければ、多層データストリームから拡張層IDフィールドを読み取り、拡張値が拡張値のドメインの第1副セット内にあるように、拡張層IDフィールドを使って拡張値を引き出し、そして、第2副フィールドの可能な値のドメインからクラスタ値のドメイン上にマッピングすることによって、クラスタ値を第2副フィールドから引き出しなさい。

【0110】

仮に、ベース層IDフィールドの第1副フィールドが、予め決められた基準を満たさないならば、多層データストリームから拡張層IDフィールドを読み取ることを控え、拡張値を拡張値のドメインの第1副セットに解体する値に設定し、そして、第2副フィールドの可能な値のドメインからクラスタ値のドメイン上にマッピングすることによって、クラスタ値を第2副フィールドから引き出しなさい。

【0111】

ここで、装置は、個々のパケットが、クラスタと拡張値を使って関連する層に索引を付ける。予め決められた基準が多層データストリームから読み取られることに注意すべきである。上で説明したように、「第1副フィールド」は、「予め決められた基準」が「仮に1つならば」、1ビットの長さおよび特にLSBである。「拡張値のドメイン」は、例えば{0、1、2、3}である。「第1副セット」は{1、2、3}である。「拡張層IDフィールドを使って」は、代わりに、ユニットのマトリクス応用の結果に第1副フィールドの値を加えている、拡張層IDフィールドに適用されたユニットのマトリクスである。「拡張値に解体する値に設定しなさい」は、「0に」設定することである。「第1副フィールドの可能な値のドメインからクラスタ値のドメイン上にマッピングすること」は、両方の場合に対して同じである。しかし、代わりに異なってもよい。マッピングはバイジェクティブである。「クラスタと拡張値を使って、個々のパケットが関連する層に索引を付ける」は、例えば、より重要でない数字を設定するために、より重要な数字および拡張値を設定することに対して、クラスタ値を使うことを含む。

【0112】

後ろの概念は、相互層予測を使って、情報量の異なるレベルに一致している異なる層で、ビデオ素材202を多層データストリーム204の中に符号化するように構成された符号器の中に直接に移送される。多層データストリームは、それぞれが異なる層の1つと関連する複数のパケット206を含む。個々のパケットが関連する層が、クラスタと拡張値によって独特に決定される。符号器は、仮に、拡張値が拡張値のドメインの第1副セット内にあるならば、多層データストリームのパケット206のそれぞれに対して、第2副フィールドを設定するためのクラスタ値を使って、予め決められた基準を満たすために第1副フィールドを設定することで、第1副フィールド308aおよび第2副フィールド308bを含むベース層IDフィールド308を多層データストリームの中に挿入するように構成されている。そして、符号器は、拡張層IDを設定するための拡張値を使って、拡張層IDを多層のデータストリームの中に挿入するように構成されている。そして、符号器は、仮に、拡張値が拡張値のドメインの第1副セットに解体する値に等しいならば、予め決められた基準を満たさないように、第2副フィールドを設定するためのクラスタ値を使って、第1副フィールドを設定することで、多層のデータストリームの中に、拡張層IDフィールドを挿入することおよびベース層IDフィールド308を挿入312することを控えるように構成されている。すなわち、符号器は、復号器が、正しい層IDを上で概説した方法で引き出して、復号側または受信側について上で概説したこれらに比較した逆マ

10

20

30

40

50

ッピングをこの終わりに使うように、データストリームを生成する。

【0113】

上の説明を要約すると、同じく、エンハンスメント層データの層識別子、クラスタの中の層識別子のグループ化およびクラスタ内の符号化依存の効率的な信号に対して、拡張メカニズムが提供される。

【0114】

上に発表された詳細は、層にされた符号化シナリオの中のエンハンスメント層のデータに適用される。一方、ベース層のデータは不変に置かれる。層にされた符号化シナリオのベース層は、ベース層とエンハンスメント層のデータパケットの共通のヘッダー構文の中の層識別子 (`n u h _ l a y e r _ i d` と比較してください) を通じて識別される。

10

【0115】

上の例では、受信者側の個々の装置はビデオデコーダである。従って動作するビデオデコーダは、復号化のための (拡張された) `l a y e r I D` によって識別されたビットストリームのパケットを選ぶことができる。しかし、装置は、代わりに、ネットワーク利用、復号器についての知識などの外部のファクターに基づいて、 (拡張された) `l a y e r I D` によって識別されるビットストリームのパケットを捨てることのできるネットワーク要素である。

【0116】

上の実施の形態の全てにおいて、多層データストリームは、別の層から予測された層相互層が、1つ以上のさらなるビュー、奥行き情報、表面反射率情報、色組み合わせ情報、空間解像度改良および `S N R` 解像度改良を追加するように、相互層予測を使って、異なる層にてその中で符号化されたビデオ素材を持つ。

20

【0117】

いくつかの面が装置の文脈において説明されたけれども、これらの面が、また、ブロックまたは装置が方法ステップまたは方法ステップの特徴に対応する対応方法の説明を表現することは明らかである。相似的に、方法ステップの文脈において説明された面は、また、対応する装置の対応するブロックまたはアイテムまたは特徴の説明を表現する。方法ステップのうちのいくつかまたは全ては、例えばマイクロプロセッサ、プログラム可能なコンピュータ、または電子回路のようなハードウェア装置によって (または使って) 実行される。いくつかの実施の形態において、最も重要な方法ステップのうちの約1つ以上は、そのような装置によって実行される。

30

【0118】

上で述べられた発明の符号化された信号は、デジタル記憶媒体に蓄えられるか、または、無線通信媒体のような通信媒体またはインターネットのような有線通信媒体で移送される。

【0119】

特定の実行要件に依存することによって、本発明の実施の形態はハードウェアまたはソフトウェアにおいて実行できる。実行は、個々の方法が実行されるようなプログラム可能なコンピュータシステムと協力し (または、協力する可能性があり)、そこに蓄えられた電子的に読み取り可能な制御信号を持つデジタル記憶媒体 (例えばフロッピー (登録商標) ディスク、DVD、ブルーレイ、CD、ROM、PROM、EPROM、EEPROM またはフラッシュメモリ) を使って実行できる。従って、デジタル記憶媒体は読み取り可能なコンピュータであってもよい。

40

【0120】

本発明に従ったいくつかの実施の形態は、ここに説明された方法のうちの1つが実行されるように、プログラム可能なコンピュータシステムと協力する可能性がある電子的に読み取り可能な制御信号を持つデータキャリアを含む。

【0121】

一般に、本発明の実施の形態は、プログラム符号を持つコンピュータプログラム製品として実行できる。コンピュータにおいてコンピュータプログラム製品が稼働するとき、プ

50

プログラム符号は方法のうちの１つを実行するために操作される。例えば、プログラム符号は機械的に読み取り可能なキャリアに蓄えられてもよい。

【０１２２】

別の実施の形態は、ここに説明された方法のうちの１つを実行するためのコンピュータプログラムを含み、機械的に読み取り可能なキャリアに蓄えられる。

【０１２３】

すなわち、本発明の方法の実施の形態は、従って、コンピュータプログラムがコンピュータにおいて稼働するとき、ここに説明された方法のうちの１つを実行するためのプログラム符号を持つコンピュータプログラムである。

【０１２４】

従って、本発明の方法の別の実施の形態は、ここに説明された方法のうちの１つを実行するためのコンピュータプログラムを含む、その上において記録されるデータキャリア（またはデジタル記憶媒体、またはコンピュータ可読媒体）である。データキャリア、デジタル記憶媒体、または記録された媒体は、一般に実在し、および／または、非移行である。

【０１２５】

従って、本発明の方法の別の実施の形態は、ここに説明された方法のうちの１つを実行するためのコンピュータプログラムを表現しているデータストリームまたは一連の信号である。データストリームまたは一連の信号は、例えばデータ通信接続（例えばインターネット）を介して転送されるように構成される。

【０１２６】

別の実施の形態は、ここに説明された方法のうちの１つを実行するように構成された、または適用された処理手段（例えばコンピュータまたはプログラム可能な論理デバイス）を含む。

【０１２７】

別の実施の形態は、ここに説明された方法のうちの１つを実行するためのコンピュータプログラムをインストールしているコンピュータを含む。

【０１２８】

本発明に従う別の実施の形態は、レシーバーに対して、ここに記述された方法のうちの１つを実行するためのコンピュータプログラムを転送する（例えば、電子的にまたは光学的に）ように構成された装置またはシステムを含む。例えば、受信器は、コンピュータ、モバイル装置、メモリー装置または同類である。例えば、装置またはシステムは、コンピュータプログラムを受信器に転送するためのファイルサーバーを含む。

【０１２９】

いくつかの実施の形態において、プログラム可能な論理デバイス（例えば、フィールドプログラム可能ゲートアレイ）は、ここに説明された方法の機能のいくつかまたは全てを実行するために使用される。いくつかの実施の形態において、フィールドプログラム可能ゲートアレイは、ここに説明された方法のうちの１つを実行するために、マイクロプロセッサと協働する。一般に、方法は、どのようなハードウェア装置によっても好ましく実行される。

【０１３０】

上に説明された実施の形態は、本発明の原則のためにのみ説明される。ここに説明された配置と詳細の部分修正と変化は、この分野における熟練した者に明白であることは理解される。従って、切迫した特許の請求項の範囲のみによって制限され、この実施の形態の説明と例示の方法によって提出された特定の詳細によって制限されない。

10

20

30

40

【 図 1 】

<code>nal_unit_header() {</code>	デスクリプタ
<code> forbidden_zero_bit</code>	<code>f(1)</code>
<code> nal_unit_type</code>	<code>u(6)</code>
<code> nuh_layer_id</code>	<code>u(6)</code>
<code> nuh_temporal_id_plus1</code>	<code>u(3)</code>
<code>}</code>	

FIG 1

【 図 2 】

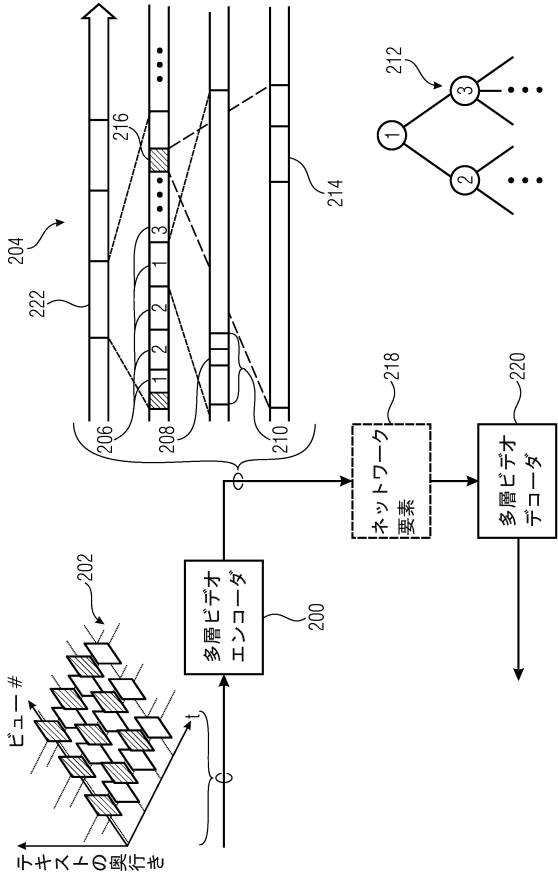


FIG 2

【 図 3 】

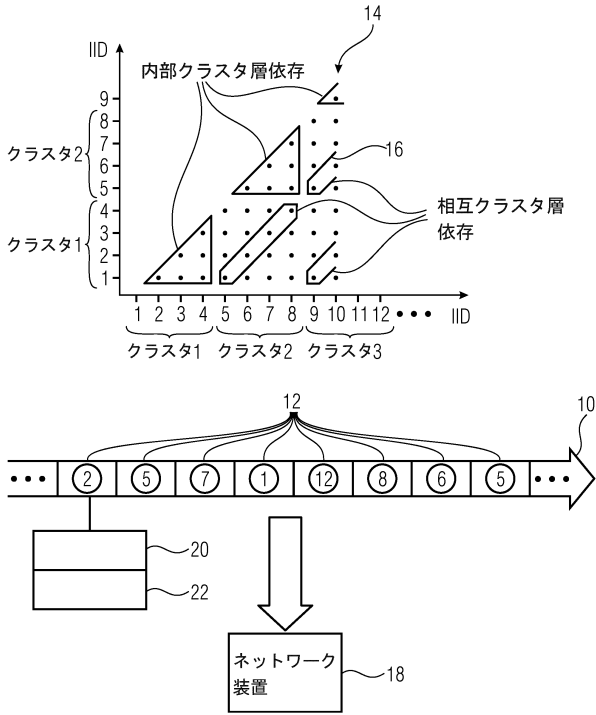


FIG 3

【 図 4 】

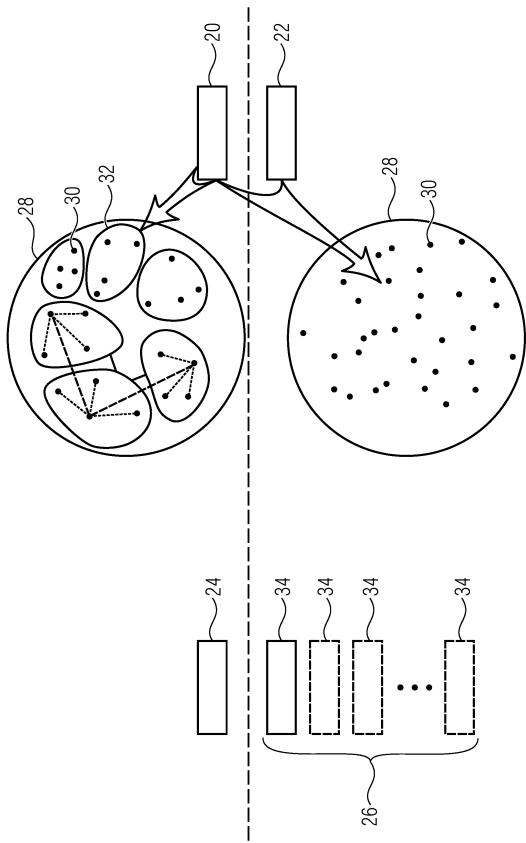


FIG 4

【 図 9 】

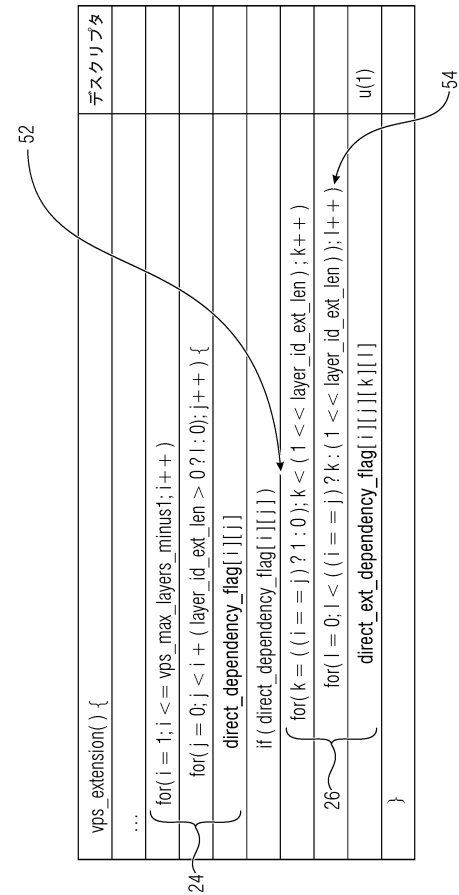


FIG 9

【 図 10 】

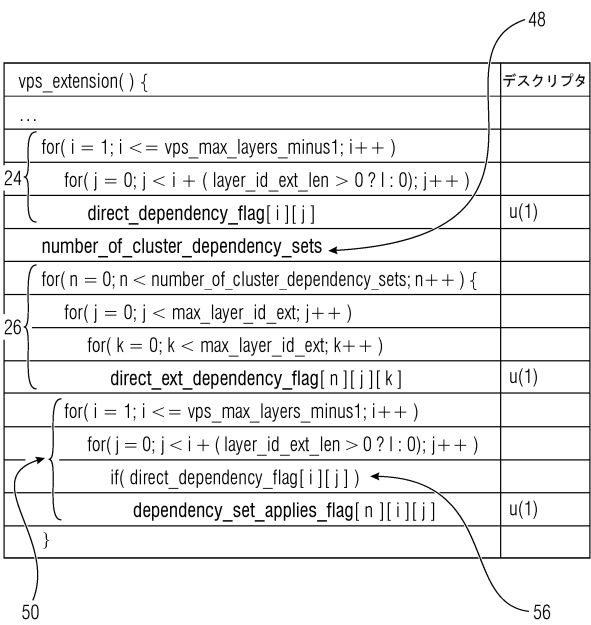


FIG 10

【 図 11 】

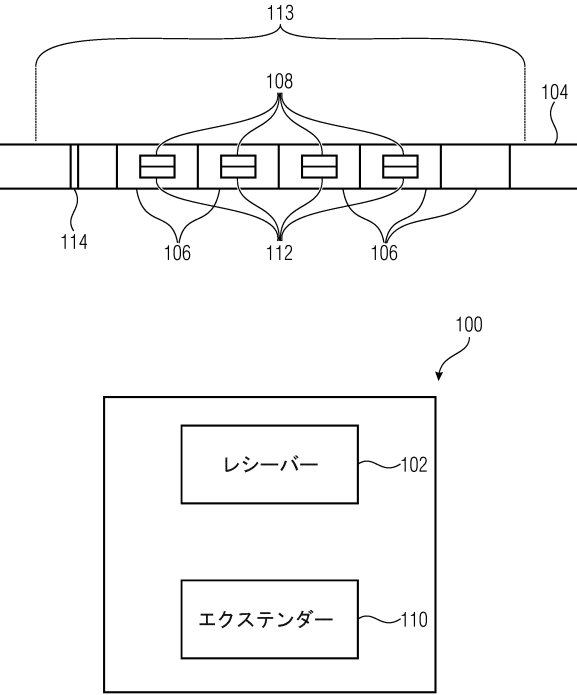


FIG 11

【 図 12 】

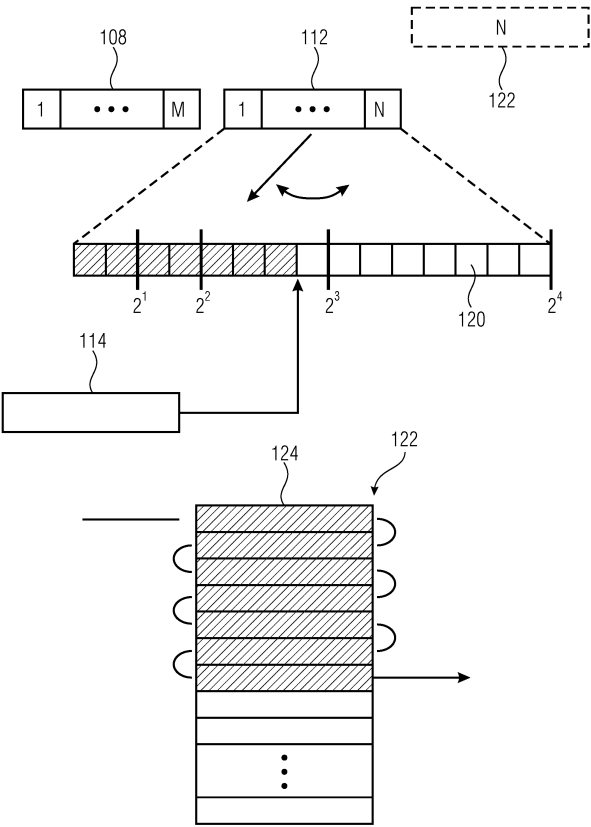


FIG 12

【図 13】

slice_segment_header() {	デスクリプタ
first_slice_segment_in_pic_flag	u(1)
...	
if(!dependent_slice_segment_flag) {	
if(nuh_layer_id > 0) {	
layer_id_ext	u(v)
}	
for(i = 0; i < (num_extra_slice_header_bits); i++)	
slice_reserved_flag[i]	u(1)
...	
}	

FIG 13

【図 14】

video_parameter_set_extension() {	デスクリプタ
...	
layer_id_ext_len	u(3)
...	
}	

FIG 14

【図 15】

video_parameter_set_extension() {	デスクリプタ
...	
max_layer_id_ext	u(3)
...	
}	

FIG 15

【図 17】

vps_extension() {	デスクリプタ
...	
Vps_nuh_layer_id_present_flag	u(1)
for(i = 1; i <= VpsMaxNumLayers - 1; i++) {	
if(vps_nuh_layer_id_present_flag)	
layer_id_in_nalu[i]	u(v)
if(!splitting_flag)	
for(j = 0; j < NumScalabilityTypes; j++)	
dimension_id[i][j]	u(v)
}	
for(i = 1; i <= VpsMaxNumLayers - 1; i++)	
for(j = 0; j < i; j++)	
direct_dependency_flag[i][j]	u(1)
for(i = 0; i < VpsMaxNumLayers - 1; i++)	
max_tid_il_ref_pics_plus1[i]	u(3)
vps_number_layer_sets_minus1	u(10)
vps_num_profile_tier_level_minus1	u(6)
for(i = 1; i <= VpsMaxNumLayers - 1; i++) {	
vps_profile_present_flag[i]	u(1)
if(!vps_profile_present_flag[i])	
profile_ref_minus1[i]	u(6)
profile_tier_level(vps_profile_present_flag[i], vps_max_sub_layers_minus1)	
}	
...	
direct_dep_type_len_minus2	ue(v)
for(i = 1; i <= VpsMaxNumLayers - 1; i++)	
for(j = 0; j < i; j++)	
if(direct_dependency_flag[i][j])	
direct_dependency_type[i][j]	u(v)
...	
}	

FIG 17

【図 16 a】

generic_sei_extension() {	デスクリプタ
...	
layer_id_ext	ue(v)
...	
}	

FIG 16A

【図 16 b】

sps_extension() {	デスクリプタ
...	
layer_id_ext	ue(v)
...	
}	

FIG 16B

【図 16 c】

pps_extension() {	デスクリプタ
...	
layer_id_ext	ue(v)
...	
}	

FIG 16C

【図 18】

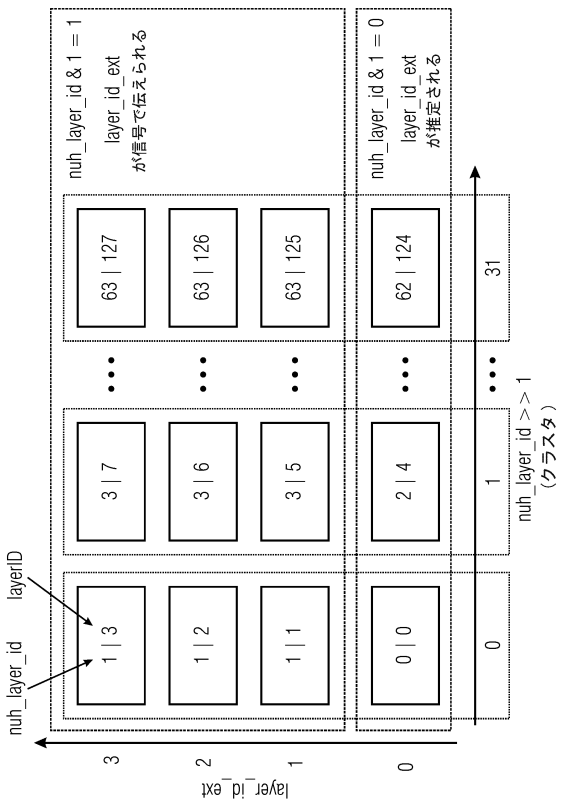


FIG 18

【図 19】

slice_segment_header() {	デスクリプタ
first_slice_segment_in_pic_flag	u(1)
...	
if(!dependent_slice_segment_flag) {	
if (nuh_layer_id & > 0) {	
layer_id_ext	u(v)
}	
for(i = 0; i < (num_extra_slice_header_bits); i++)	
slice_reserved_flag[i]	u(1)
...	
}	

FIG 19

【図 20 b】

slice_segment_header() {	デスクリプタ
first_slice_segment_in_pic_flag	u(1)
...	
if(!dependent_slice_segment_flag) {	
maxLayerIdExtOffset = (1 << add_layer_id_ext_len) - 1	
layerIdExtOffset = nuh_layer_id & maxLayerIdExtOffset	
if (layerIdExtOffset == maxLayerIdExtOffset) {	
layer_id_ext	u(v)
}	
for(i = 0; i < (num_extra_slice_header_bits); i++)	
slice_reserved_flag[i]	u(1)
...	
}	

FIG 20B

【図 20 a】

video_parameter_set_extension() {	デスクリプタ
...	
add_layer_id_ext_len	u(3)
...	
}	

FIG 20A

【図 21】

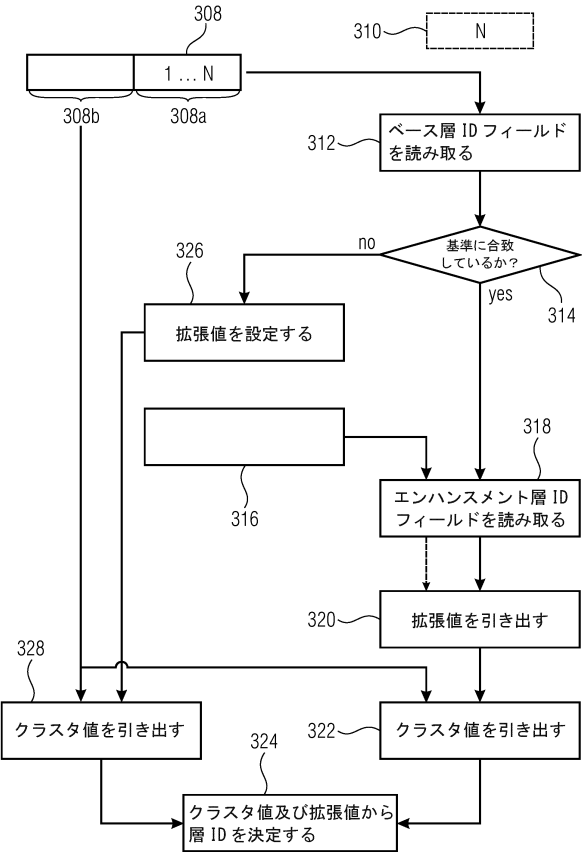


FIG 21

フロントページの続き

- (72)発明者 マルペ デトレフ
ドイツ連邦共和国 1 2 1 6 1 ベルリン ズュートヴェストコルゾ 7 0
- (72)発明者 スクービン ローベルト
ドイツ連邦共和国 1 0 5 5 5 ベルリン シュレースヴィガー ウーファー 5
- (72)発明者 サンチェス デ ラ フエンテ ヤーゴ
ドイツ連邦共和国 1 0 2 4 3 ベルリン ヴァルシャウアー シュトラーセ 6 7
- (72)発明者 テヒ ゲルハルト
ドイツ連邦共和国 1 0 5 5 7 ベルリン メランヒトンシュトラーセ 1 3

合議体

審判長 清水 正一

審判官 榎本 剛

審判官 渡辺 努

- (56)参考文献 米国特許出願公開第2012/0183077(US, A1)
Karsten Suehring et al., "MV-HEVC/SHVC HLS:
Extending the supported number of layers", J
oint Collaborative Team on 3D Video Coding
Extensions of ITU-T SG 16 WP 3 and ISO/IEC
JTC 1/SC 29/WG 11, JCT3V-D0211_v1, 2013.04.26
, p. 1 - 7

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