

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第5116175号  
(P5116175)

(45) 発行日 平成25年1月9日(2013.1.9)

(24) 登録日 平成24年10月26日(2012.10.26)

(51) Int.Cl.

HO4N 7/32 (2006.01)

F 1

HO4N 7/137

Z

請求項の数 5 (全 31 頁)

(21) 出願番号	特願2009-532450 (P2009-532450)
(86) (22) 出願日	平成19年10月12日 (2007.10.12)
(65) 公表番号	特表2010-507277 (P2010-507277A)
(43) 公表日	平成22年3月4日 (2010.3.4)
(86) 國際出願番号	PCT/US2007/021902
(87) 國際公開番号	W02008/048515
(87) 國際公開日	平成20年4月24日 (2008.4.24)
審査請求日	平成22年10月8日 (2010.10.8)
(31) 優先権主張番号	60/851,522
(32) 優先日	平成18年10月13日 (2006.10.13)
(33) 優先権主張国	米国(US)
(31) 優先権主張番号	60/851,589
(32) 優先日	平成18年10月13日 (2006.10.13)
(33) 優先権主張国	米国(US)

(73) 特許権者	501263810 トムソン ライセンシング Thomson Licensing フランス国, 92130 イッシー レ ムーリノー, ル ジヤンヌ ダルク, 1-5 1-5, rue Jeanne d' A rc, 92130 ISSY LES MOULINEAUX, France
(74) 代理人	100115864 弁理士 木越 力
(74) 代理人	100121175 弁理士 石井 たかし

最終頁に続く

(54) 【発明の名称】多視点ビデオ符号化を含む参照ピクチャの管理方法

## (57) 【特許請求の範囲】

## 【請求項 1】

多視点ビデオ符号化において使用される第1の参照ピクチャのメモリの管理方法であつて、

第1の参照ピクチャをメモリに保存するステップであつて、前記第1の参照ピクチャが、メモリ・ステータスおよびビューに関連づけられている、該ステップと、

保存された前記第1の参照ピクチャの前記メモリ・ステータスに影響を及ぼす情報とともにビデオ・ピクチャをコーディングするステップであつて、前記コーディングされるビデオ・ピクチャが、前記第1の参照ピクチャに関連づけられた前記ビューとは異なるビューに関連づけられている、該ステップと、

保存された前記第1の参照ピクチャに関連づけられたメモリ・ステータスを、長期参照ピクチャ、短期参照ピクチャ、および、参照に使用されない、のうちから選択されるメモリ・ステータスから、長期参照ピクチャ、短期参照ピクチャ、および参照に使用されない、のうちから選択されるメモリ・ステータスにメモリ管理変更コマンドを介して指定するステップであつて、参照に使用されないに指定する場合には、前記第1の参照ピクチャが前記メモリから削除される、前記ステップを含み、

保存された第2の参照ピクチャのメモリ・ステータスが、前記第1の参照ピクチャのビューとは異なる第2のビューに関連づけられており、前記ビデオ・ピクチャのビューは、前記メモリ管理変更コマンドが前記第1の参照ピクチャにのみ影響を及ぼすように、コーディングされる、前記方法。

10

20

## 【請求項 2】

前記第1及び第2の参照ピクチャが、最初にH.264ベースのコーディングを使用してコーディングされ、前記メモリ・ステータスの変更が、多視点コーディングの最中に実行される、請求項1に記載の方法。

## 【請求項 3】

前記第1及び第2の参照ピクチャが、コーディングされ、前記メモリ・ステータスの前記変更が、時間的コーディングおよびビュー間コーディングの両方を行うビデオ・コーディングにおいて実行される、請求項1に記載の方法。

## 【請求項 4】

マーク付けモード・シンタックス要素フラグが、現在コーディングされている前記ビデオ・ピクチャの参照マーク付けモードを選択するために呼び出される、請求項1～3のいずれか1項に記載の方法。 10

## 【請求項 5】

請求項1～4のいずれか1項に記載の方法を実行するコーディング装置。

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

## (関連出願の相互参照)

本出願は、参照によりその全体が本明細書に組み込まれる、2006年10月13日に出願された、米国仮出願第60/851,522号、および同様に参照によりその全体が本明細書に組み込まれる、2006年10月13日に出願された、米国仮出願第60/851,589号の利益を主張する。 20

## 【0002】

本発明は、動画像の分野に関し、特に、多視点ビデオ符号化に関連する動画像のメモリ保持の問題に関する。

## 【背景技術】

## 【0003】

多くのフレーム間符号化システムは、参照ピクチャを使用し、そのような参照ピクチャの使用は、符号化されたビット・ストリームのサイズを縮小するのに役立つ。このタイプの結果は、符号化効率がフレーム内符号化技法を単独で単に使用するよりも良好になるということである。したがって、多くの符号化規格は、一連の動画からのビット・ストリームを符号化するために、フレーム内符号化技法およびフレーム間符号化技法の両方を含む。当技術分野で知られているように、異なるタイプの参照ピクチャが、符号化規格のために使用される。例えば、ピクチャそれ自体の内部の要素(イントラ・フレーム)だけを使用することによって符号化されるIピクチャ、ピクチャそれ自体の内部からの要素および/または2つの先行する参照ピクチャからの要素(インター・フレーム)を使用することによって符号化されるBピクチャ、ならびにピクチャそれ自体の内部からの要素および/または1つの先行する参照ピクチャからの要素(インター・フレーム)を使用することによって符号化されるPピクチャなどがある。BピクチャおよびPピクチャはともに、複数の参照ピクチャを使用することができるが、これらのタイプ双方のピクチャの間の相違は、Bピクチャは、ブロック当たり最大2つの動き補償予測信号を用いるインター予測(intra prediction)の使用を可能にするが、Pピクチャは、予測ブロック当たりただ1つの予測器の使用を可能にするに過ぎないことである。 30

## 【0004】

したがって、BピクチャまたはPピクチャが符号化されて生成される場合や復号される場合、そのようなピクチャが、適切に符号化されまたは復号動作中に復元され得るよう、そのようなピクチャは他の参照フレームに依存する。符号化/復号システムは、他のピクチャが参照ピクチャを考慮して符号化または復号されている間、参照ピクチャを保存できるように、何らかのタイプのメモリ・ロケーションを提供すべきである。明らかに、しばらくすると、参照ピクチャは符号化処理のために使用することができない。将来の符号 40

50

化処理において、符号化されるピクチャがその参照ピクチャをもはや使用しなくなるためである。

【0005】

すべての参照ピクチャを記憶デバイス内に永続的に保存することもできるが、そのようなソリューションは、メモリ資源を非効率的に使用することになる。したがって、参照ピクチャを保存するメモリ・デバイスを操作する場合に、そのような参照ピクチャに必要とされるスペースを（不要な参照ピクチャを廃棄することによって）削減する助けとなるよう、当技術分野で知られた、先入れ先出し（FIFO）または後入れ先出し（LIFO）メモリ操作を使用するようなメモリ技法が使用される。しかしながら、符号化および/または復号されるピクチャが時間的相互関係およびビュー相互関係をともに有する、多視点符号化システムの使用を考慮する場合、そのようなメモリ操作は望ましくない結果を生じことがある。すなわち、多視点符号化システムは、各ビューがそれぞれの物体／場面の異なる視点からの眺めを表す、複数のビューの動画を有する態様を導入する。今日、参照ピクチャは、2つの異なるビューに関連付けられたピクチャの符号化処理または復号処理において使用される。したがって、単純なメモリ技法はそのような環境において使用することはできない。

【先行技術文献】

【非特許文献】

【0006】

【非特許文献1】A. Vetro、Y. Su、H. Kimata、A. Smolic、「Joint Multiview Video Model (JMVM) 1.0」、JVT-T208.doc、Klagenfurt、Austria、2006年7月  
 【非特許文献2】G. Sullivan、T. Wiegand、A. Luthra、「Draft of Version 4 of H.264/AVC (ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 part 10) Advanced Video Coding)」、Palma de Mallorca, ES 18-22、2004年10月

【発明の概要】

【0007】

本発明の原理は、従来技術の上記および他の難点および不都合に対処するものであり、ビデオ符号化のための動き推定予測器として利用可能な動き情報を再使用するための方法および装置に関する。

【0008】

本原理の一態様によれば、復号器によって復号されるピクチャからの情報を考慮して、メモリ・デバイス内に保存された参照ピクチャに対してメモリ管理操作を実行する、コーダが提供され、そのような情報は、その参照ピクチャに関連するビュー情報に関係する。

【0009】

本原理の上記および他の態様、特徴、および利点は、添付の図面と関連させて読まれる、例示的な実施形態の以下の詳細な説明から明らかになろう。

【0010】

本原理は、以下の例示的な図面によって、より良く理解することができよう。

【図面の簡単な説明】

【0011】

【図1】異なる時点におけるビデオ・ピクチャ・ビューの多視点符号化の例示的な一実施形態であって、そのようなビデオ・ピクチャが、図に示された方法で参照ピクチャを使用してコーディングされる、一実施形態を示す図である。

【図2】本発明の原理による、ビデオ・コーダの例示的な一実施形態を示す図である。

【図3】本発明の原理に従って使用される、シンタックス要素 `dec_ref_pic_marking()` の疑似コードの一実施形態を示す図である。

【図4】本発明の原理に従って使用される、シンタックス要素 `seq_params`

10

20

30

40

50

`r_set_mv_c_extension()` の疑似コードを示す図である。

【図5】本発明の原理に従って使用される、シンタックス要素`dec_ref_pic_marking_mv_c_extension()` の疑似コードの一実施形態を示す図である。

【図6】本発明の原理に従って使用される、サンプル・ピクチャ・ヘッダの一実施形態を示す図である。

【図6 A】図6に示される図6の一部を示す図である。

【図6 B】図6に示される図6の他の一部を示す図である。

【図7】本発明の原理に従って使用される、シンタックス要素`dec_ref_pic_marking_mv_c_extension()` の疑似コードの一実施形態を示す図である。

10

【図8】本発明の原理に従って使用される、サンプル・ピクチャ・ヘッダの一実施形態を示す図である。

【図8 A】図8に示される図8の一部を示す図である。

【図8 B】図8に示される図8の他の一部を示す図である。

【図9】本発明の原理に従って使用される、シンタックス要素`dec_ref_pic_marking_mv_c_extension()` の疑似コードの一実施形態を示す図である。

【図10】本発明の原理に従って使用される、シンタックス要素`dec_ref_pic_marking_mv_c_extension()` の疑似コードの一実施形態を示す図である。

20

【図11】本発明の原理による、ピクチャのマーキング方法の一実施形態のブロック図である。

【発明を実施するための形態】

【0012】

本発明の原理は、任意のフレーム内およびフレーム間ベースの符号化規格に適用することができる。本明細書の全体で使用される「ピクチャ」という用語は、当技術分野において、「フレーム」、「フィールド」、および「スライス」として、ならびに「ピクチャ」という用語自体でも知られ得る、様々な形態のビデオ画像情報を説明するための総称的用語として使用される。

30

【0013】

また、本発明の説明では、C言語タイプの書式を使用する様々なコマンド（シンタックス要素）が、そのようなコマンド中のディスクリプタについて以下の用語法を使用する図において詳述される。

`u ( n )` : nビットを使用する符号なし整数。シンタックス表内でnが「v」である場合、ビットの数は、他のシンタックス要素の値に依存する方法で変化する。このディスクリプタのための構文解析プロセスは、最上位ビットが最初に書かれる、符号なし整数のバイナリ表現として解釈される、関数`read_bits ( n )`の戻り値によって規定される。

`ue ( v )` : 左ビットが最初の、符号なし整数の指数ゴロム符号化 (Exp-Golomb) シンタックス要素。

40

`se ( v )` : 左ビットが最初の、符号付き整数の指数ゴロム符号化シンタックス要素。  
C : シンタックス要素が適用されるカテゴリを、すなわち、特定のフィールドがどのレベルに適用されるべきかを表す。

【0014】

本説明は、本原理を説明する。したがって、本明細書で明示的に説明され、または示されていなくても、本原理を実施し、その主旨および範囲内に含まれる様々な構成を、当業者であれば考案できることが理解されよう。

【0015】

本明細書で述べられるすべての例および条件付きの言葉は、教示を目的とするもので、

50

本原理と当技術分野を発展させるために発明者（ら）が寄与した概念とを読者が理解する助けとなるように意図されており、そのような具体的に述べられた例および条件に限定されると解釈されるべきではない。

【0016】

さらに、本明細書で本原理の原理、態様、および実施形態を述べるすべての言表は、それらの具体的な例と同様に、それらの構造的均等物および機能的均等物の両方を包含することが意図されている。加えて、そのような均等物は、現在知られている均等物ばかりではなく、将来開発される均等物も含むことが、すなわち、構造に関わらず同一機能を実行する、開発されたいかなる要素をも含むことが意図されている。

【0017】

したがって、例えば、本明細書で提示されるブロック図は、本原理を実施する説明的な回路の概念図を表すことが、当業者であれば理解されよう。同様に、いかなるフローチャート、フロー図、状態遷移図、および疑似コードなども、実際にはコンピュータ可読媒体内で提示され、そのため、コンピュータまたはプロセッサが明示的に示されているかどうかに関わらず、そのようなコンピュータまたはプロセッサによって実行され得る、様々なプロセスを表すことが理解されよう。

【0018】

図面に示された様々な要素の機能は、専用ハードウェアの使用を通してだけでなく、適切なソフトウェアと関連する、ソフトウェアを実行可能なハードウェアの使用を通して提供することができる。プロセッサによって提供される場合、機能は、単一の専用プロセッサによって、単一の共用プロセッサによって、またはその中のいくつかは共用されてよい複数の個別プロセッサによって提供することができる。さらに、「プロセッサ」または「コントローラ」という用語の明示的な使用は、ソフトウェアを実行可能なハードウェアを排他的に指示すると解釈されるべきではなく、デジタル信号プロセッサ（「DSP」）ハードウェア、ソフトウェアを保存するためのリード・オンリ・メモリ（「ROM」）、ランダム・アクセス・メモリ（「RAM」）、および不揮発性ストレージを、これらに限定することなく、暗黙的に含むことができる。

【0019】

他の従来のハードウェアおよび/またはカスタム・ハードウェアも含むことができる。同様に、図面に示されたスイッチはいずれも、概念的なものに過ぎない。それらの機能は、プログラム・ロジックの動作を通して、専用ロジックを通して、またはプログラム制御と専用ロジックの相互作用を通して実施することができ、具体的な技法は、文脈からより具体的に理解されるように、実施者によって選択可能である。

【0020】

本願の特許請求の範囲では、指定された機能を実行する手段として表された要素はいずれも、例えば、a) その機能を実行する回路要素の組合せ、またはb) 任意の形態のソフトウェアであって、したがって、ファームウェアまたはマイクロコードなどを含み、そのソフトウェアを実行するための適切な回路と組み合わされて当該機能を実行するソフトウェアを含む、その機能を実行する任意の方法を包含することが意図されている。そのような特許請求の範囲によって確定される本原理は、列挙された様々な手段によって提供される機能が、特許請求の範囲が要請する方法で組み合わされ、一緒にされるという事実に存している。したがって、それらの機能を提供できる手段はいずれも、本明細書で示される手段と等価であると見なされる。

【0021】

本明細書における本原理の「一実施形態」または「実施形態」に対する言及は、その実施形態に関連して説明された特定の機能、構造、および特徴などが、本原理の少なくとも1つの実施形態に含まれることを意味する。したがって、本明細書の様々な場所で見られる「一実施形態では」または「実施形態では」という句の出現は、必ずしもすべてが同一の実施形態に言及しているわけではない。

【0022】

10

20

30

40

50

図1は、多視点符号化システムにおいて使用される、参照ピクチャ構造の例示的な一実施形態を表している。具体的には、示されている構造は、多視点符号化(MVC)スキームによる、8つの異なるビュー(view)(S0~S7)の多時点(T0~T100)にわたる使用に関する。この多視点符号化(MVC)スキームは、A.Vetro、Y.Su、H.Kimata、A.Smolica、「Joint Multiview Video Model (JMVM) 1.0」、JVT-T208.doc、Klagenfurt、Austria、2006年7月において提案されたものである。この多視点符号化規格は、AVC(Advanced Video Coding)規格(G.Sullivan、T.Wiegand、A.Luthra、「Draft of Version 4 of H.264/AVC (ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 part 10) Advanced Video Coding)」、Palma de Mallorca、ES 18-22、2004年10月)における符号化に基づいている。両規格の大きな相違は、AVCは多視点ピクチャのコーディングに対処しないが、MVCは対処することである。

#### 【0023】

図1をまた参照すると、例えば、T1におけるビューS1に関連するピクチャをコーディングする場合、コーディングされるピクチャは、同じビュー(T0におけるS1およびT2におけるS1)からのピクチャ(参照ピクチャ)に関係し、またコーディングされるピクチャは、異なるビュー(T1におけるS0およびT1におけるS2)からのピクチャに関係することが分かる。したがって、S1、T1に関連するピクチャをコーディングする場合、参照ピクチャ(T0におけるS1、T2におけるS1、T1におけるS0、およびT1におけるS2)を、ハードウェア、ソフトウェア、またはそれらの組合せで実施できる、バッファ、レジスタ、およびRAMなどの、メモリ・デバイスに保持することは理に適っている。しかし、そのような参照ピクチャは、T1におけるS1のピクチャのために使用されるのとは異なる参照ピクチャの使用に依存する、T98におけるS7の符号化を考える場合、有用ではない。

#### 【0024】

コーディング操作のためのバッファの効果的なメモリ管理についての1つのソリューションが、AVCビデオ規格に関連する復号済ピクチャ・バッファ(DPB:decode picture buffer)の使用において開示される。簡略版である図2のブロック図200において、コーダ205と、コーディング・バッファ210と、復号済ピクチャ・バッファ(DPB)215の間の動作が示されている。コーディング動作(符号化または復号)の最中、コーダ205によって現在コーディングされているピクチャは、コーディング・バッファ210内に存在するが、先にコーディングされた参照ピクチャは、復号済ピクチャ・バッファ215内に保存されている。AVCは、復号済ピクチャ・バッファ215内の参照ピクチャがどのように保持されるべきかをコーダ205が規定することを可能にする、メモリ管理制御操作(MMCO:memory management control operation)として知られるコマンドの使用を開示している。すなわち、ピクチャが符号化される場合、そのようなピクチャより前に到来した参照ピクチャを用いて何が行われるべきかを規定するために、そのようなMMCOが、これから符号化されるピクチャのヘッダに入力される。この操作は、「マーキング(mark ing)」として知られている。その後、これらのコマンドは、復号済ピクチャ・バッファ215内に存在する参照ピクチャを用いて何が行われるべきかを決定するために、将来にコーダ205によって使用されることができる。ピクチャという用語は、様々な要素ビデオ情報を表すために使用されるが、そのような参照ピクチャが同じピクチャからのスライスを「参照ピクチャ」として使用できる場合、AVCはスライスという用語を使用するようにしており、またピクチャがどのように細分できるかに関わらず、本発明の原理が適用されることに留意されたい。

#### 【0025】

10

20

30

40

50

図3は、MMC0コマンドを実施するために使用される、AVCにおけるコマンド(`dec_ref_pic_marking`)を表している。具体的には、MMC0コマンドを利用する場合、参照ピクチャは、短期参照ピクチャ、長期参照ピクチャとしてマーキングされ、またはピクチャは、参照ピクチャではないとしてマーキングされる(その場合、参照ピクチャは、メモリが必要とされる場合には廃棄される)。参照ピクチャのステータスは、より多くのピクチャがコーディングされるにつれて変更されてよく、例えば、1つのピクチャがコーディングされるピクチャである場合に短期として指定された参照ピクチャは、第2のピクチャがコーディングされる場合には長期参照ピクチャとして識別されることができる。

【0026】

10

図3は、ピクチャ・ヘッダ(ライス・ヘッダ)のマーキングを実行するために2つの異なるモードの間で使用される、`adaptive_ref_pic_marking_mode_flag`として知られるコマンド・フラグも提示している。フラグが「0」に設定された場合、短期参照ピクチャのための FIFO 機構を提供する、スライディング・ウィンドウ参照マーキング・モードが起動される。フラグが「1」に設定された場合、参照ピクチャを「unused for reference」としてマーキングするため、および参照ピクチャに長期フレーム・インデックスを割り当てるためなどのシンタックス要素を提供する、適応参照ピクチャ・マーキング・モードが起動される。AVCにおいて使用される、MMC0コマンドを介した、参照フレームのための様々な割り当てが、以下の表1に示されている。

【0027】

20

【表1】

memory_management_control_operation	メモリ管理制御操作
0	memory_management_control_operation シンタックス要素のループを終了する
1	短期参照ピクチャを「unused for reference (参照に使用されない)」としてマーキングする
2	長期参照ピクチャを「unused for reference (参照に使用されない)」としてマーキングする
3	短期参照ピクチャを「used for long-term reference (長期参照に使用される)」としてマーキングし、それに長期フレーム・インデックスを割り当てる
4	最大長期フレーム・インデックスを指定し、最大値よりも大きな長期フレーム・インデックスを有するすべての長期参照ピクチャを「unused for reference (参照に使用されない)」としてマーキングする
5	すべての参照ピクチャを「unused for reference (参照に使用されない)」としてマーキングし、MaxLongTermFrameIdx 変数に「no long-term frame indices (長期フレーム・インデックスでない)」を設定する
6	現ピクチャを「used for long-term reference (長期参照に使用される)」としてマーキングし、それに長期フレーム・インデックスを割り当てる

表1

10

20

30

【0028】

AVCの設計に関する1つの問題は、ピクチャが、(ピクチャの系列内における)ピクチャの実際のコーディング順序を表す、それぞれのフレーム番号(frame\_num)値によって、およびピクチャが表示される順序である、ピクチャのそれぞれのピクチャ順序カウント(POC:picture order count)によって、識別できることである。しかし、AVCは単に単一のビューを考慮するだけなのに対して、MVCにおいては複数のビューが考慮されなければならないので、MVCはAVCより複雑である。したがって、MVCにおいては、特定のピクチャを特定のビューに関連づけるために、付加的な値ビューid(view\_id)が使用される。

40

【0029】

したがって、AVCのMMCOを、MVCからのview\_idの使用と組み合わせる場合、従来技術のMMCOの現行の使用法は、ユーザが、同じview\_idのピクチャに従って、MMCOを供給することを可能にするに過ぎない。すなわち、コーディングされるピクチャは、同じview\_idタイプの他のピクチャを参照できるに過ぎない(ビュー1タイプのピクチャは、他のビュー1タイプのピクチャにMMCOコマンドを供給できるに過ぎない)。MMCOコマンドの現行の使用法を用いてこれら異なるビューのすべてを追跡し続けなければならない場合、DPB操作のためのメモリ管理の効率的な使用の

50

妨げとなる。

【0030】

具体的には、図4は、付加的なシンタックスが、（クロスピュー参照を通知するために使用される）S P Sにおいて追加された、現行のM V Cを表している。追加されるシンタックスは、以下で説明される方法で、アンカ・ピクチャ（a n c h o r \_ p i c t u r e）（すなわち、Iピクチャ）および非アンカ・ピクチャ（n o n - a n c h o r \_ p i c t u r e）のために使用される、クロスピュー参照を指示するために使用される。

【0031】

アンカ・ピクチャは典型的には、すべてのスライスが、同じピクチャ順序カウントを有するスライス、すなわち、他のビューのスライスのみを参照し、現ビューのスライスを参照しない、コーディングされるピクチャを表すことに留意されたい。そのようなピクチャは、a n c h o r \_ p i c \_ f l a g を1に設定することによって通知される。アンカ・ピクチャを復号した後、表示順序で続いてコーディングされるピクチャのすべては、アンカ・ピクチャに先立って復号された他の任意のピクチャからのインター予測を使用する中で、復号されることが可能である。1つのビュー内のピクチャがアンカ・ピクチャである場合、他のビューの同じ時間インデックス内のすべてのピクチャも、アンカ・ピクチャとして知られる。

【0032】

以下の手順は、現ビューとは異なるビューからの参照ピクチャを参照予測リスト内に配置するために実行される。

- 0からn u m \_ m u l t i v i e w \_ r e f \_ f o r \_ l i s t X - 1までの「I」の各値について、
- 現ピクチャと時間的に並んだビューr e f e r e n c \_ v i e w \_ f o r \_ l i s t \_ X [ i ]からの再構成済みピクチャが取得され、復号済ピクチャ・バッファ（D P B）内に挿入される。
- そのピクチャのインデックスが、R e f P i c L i s t Xの次の空スロット内に挿入される。

【0033】

この詳述された実施において、M M C Oコマンドは、個々のビューとだけ直接関連し、他のビュー内のピクチャをマーキングすることはできない。その直接の結果として、クロスピュー参照ピクチャは、ビット・ストリームのより後方のそれ自体のビュー内のピクチャによって、u n u s e d \_ f o r \_ r e f e r e n c e（参照に使用されない）としてマーキングされ得るに過ぎないので、（上で示されたように）必要以上に長くD P B内に留まることができる。例えば、図1を参照すると、ビューS 0内のT 0～T 11までのピクチャは、ビューS 1、S 2のためだけに必要とされ、その後、u s e d \_ f o r \_ r e f e r e n c e（参照に使用される）としてマーキングされる。したがって、そのようなピクチャを保存するD P Bは、大きな記憶領域を必要とする。したがって、ピクチャがどのビューに関連するかを考慮せずにD P Bを消去する唯一の方法においては、新しいグループ・オブ・ピクチャ（G O P）の開始に関連するピクチャ、または即時復号リフレッシュ（I D R : i n s t a n t a n e o u s \_ d e c o d i n g \_ r e f r e s h ）ピクチャが、参照ピクチャのD P Bを完全に消去するように指示する。

【0034】

したがって、本発明は、ピクチャがコーディングされる場合、そのようなピクチャがビュー（現在コーディングされているピクチャとは同じでないビュー）間にわたる参照ピクチャをどのように考慮するかに関する情報を含むことを意味する、クロスピューに適用できるM M C Oを提供することによって、D P B問題へのソリューションを提案する。

【0035】

本発明のいくつかの実施形態は、A V C規格を考慮して提示され、新しい高レベルのシンタックス要素が定義され、説明されるが、本原理は多視点ピクチャを使用する他のコーディング規格にも適用されることを理解されたい。

10

20

30

40

50

## 【0036】

図5で提示される一実施形態では、ビュー間にわたるピクチャをマーキングするために、新しいシンタックス要素 `dec_ref_pic_marking_mvcextension()` が使用される。この関数は、コーディングされるピクチャのピクチャ・ヘッダ・シンタックスを表す、図6で提示される対応する `slice_header_mvcextension()` 関数（具体的には、このコマンドは AVC で示されるスライス・ヘッダから改造される）から呼び出される。

## 【0037】

この新しいシンタックスは、現ビュー以外のビュー内にあるピクチャをマーキングするためだけに使用されるので、システムが同じビュー内のピクチャをマーキングすることを可能にするオプションを提供することも検討されなければならない。同じビュー内のピクチャのマーキングは、新しいマーキング・プロセスの後、AVC 互換関数 `dec_ref_pic_marking`（図5を参照）を呼び出すことによって可能にされる。そのような関数は、MVC ベースのマーキングの起動の前または後に、呼び出されてよいことに留意されたい。

10

## 【0038】

多視点システムは最初 AVC 規格では対処されなかったので、AVC シンタックスは単一のビューのみを仮定するという、付加的な制約が、MVC 用の AVC ベース `dec_ref_pic_marking()` シンタックスに課される。したがって、AVC シンタックスは、コーディングされる現ピクチャが属するビューにのみ適用されなければならない。

20

## 【0039】

図5をまた参照すると、以下で定義されるいくつかの付加的なシンタックスが追加されている。

## 【0040】

現在コーディングされているピクチャの参照マーキング・モードを選択するために、`mvc_adaptive_ref_pic_marking_mode_flag` が使用される。フラグは、「0」では、スライディング・ウィンドウ参照ピクチャ・マーキング・モードを表し、短期参照ピクチャが FIFO に基づいて DPB 内に割り当てられる。フラグは、「1」では、適応参照ピクチャ・マーキング・モードを表し、現在コーディングされているピクチャに関連するビュー以外のビュー内の参照ピクチャをマーキングするために要素が提供されることができる。他のビュー内の参照ピクチャのそのようなステータスは、「`unused for reference`（参照に使用されない）」および「`long-term frame indices`（長期フレーム・インデックス）」を含む。

30

## 【0041】

現在「`used for long-term reference`（長期参照に使用される）」としてマーキングされているピクチャ（フレーム、相補的フィールド対（`complementary field pair`）、および非フィールド・フィールド（`non-field field`））の数が、`Max(Num_ref_frames, 1)` に等しい場合、フラグは 1 に等しくなる。

40

## 【0042】

`memory_management_control_operation` は、コードによる参照ピクチャ・マーキング操作に影響を及ぼすために適用される制御操作（MMCO）を規定する。`memory_management_control_operation` シンタックス要素には、制御操作の値によって指定される操作にとって必要なデータが後続する。多視点のための MMCO に関する値および制御操作は、以下の表 2 で規定される。`memory_management_control_operation` シンタックス要素は、そのようなコマンドがピクチャ・ヘッダ（例えば、スライス・ヘッダ）内に出現した順序で、コーディング処理によって処理され、各 MMCO に対して述

50

べられたセマンティック制約は、その個々のMMC0が処理される特定の位置において適用される。

【0043】

【表2】

memory_management_control_operation	メモリ管理制御操作
0	memory_management_control_operation シンタックス要素のループを終了する
1	それ自体以外のビュー内の短期参照ピクチャを「unused for reference (参照に使用されない)」としてマーキングする
2	それ自体以外のビュー内の長期参照ピクチャを「unused for reference (参照に使用されない)」としてマーキングする
3	それ自体以外のビュー内の短期参照ピクチャを「used for long-term reference (長期参照に使用される)」としてマーキングし、それに長期フレーム・インデックスを割り当てる
4	最大長期フレーム・インデックスを指定し、最大値よりも大きな長期フレーム・インデックスを有するすべての長期参照ピクチャを「unused for reference (参照に使用されない)」としてマーキングする
5	ビュー内のすべての参照ピクチャを「unused for reference (参照に使用されない)」としてマーキングする
6	それ自体以外のすべてのビュー内のすべての参照ピクチャを「unused for reference (参照に使用されない)」としてマーキングし、MaxLongTermFrameIdx 変数に「no long-term frame indices (長期フレーム・インデックスでない)」を設定する
7	それ自体以外のビュー内の長期参照ピクチャを「used for short-term reference (短期参照に使用される)」としてマーキングする

表2

【0044】

memory\_management\_control\_operation は、memory\_management\_control\_operation がコーディング・プロセスによって処理される場合に、指定された参照ピクチャが「used for short-term reference (短期参照に使用される)」としてマーキングされない限り、ピクチャ・ヘッダ (例えば、スライス・ヘッダ) 内において、1 に等しくならない。

【0045】

memory\_management\_control\_operation は、memory\_management\_control\_operation が復号プロセスによって処理される場合に、指定された長期ピクチャ番号が「used for lon

10

20

30

40

50

g - term reference」としてマーキングされた参照ピクチャを指示しない限り、スライス・ヘッダ内において、2に等しくならない。

【0046】

memory\_management\_control\_operationは、memory\_management\_control\_operationが復号プロセスによって処理される場合に、指定された参照ピクチャが「used for short-term reference」としてマーキングされない限り、スライス・ヘッダ内において、3に等しくならない。

【0047】

memory\_management\_control\_operationは、memory\_management\_control\_operationが復号プロセスによって処理される場合に、変数MaxLongTermFrameIdxの値が「no long-term frame indices」に等しい場合、3、5、または6に等しくならない。

【0048】

4に等しい2つ以上のmemory\_management\_control\_operationは、ピクチャ・ヘッダ内に存在しない。

【0049】

5に等しい2つ以上のmemory\_management\_control\_operationは、ピクチャ・ヘッダ内に存在しない。

【0050】

6に等しい2つ以上のmemory\_management\_control\_operationは、ピクチャ・ヘッダ内に存在しない。

【0051】

フィールドを復号する際に、短期参照フレームの一部または短期相補的参照フィールド対の一部であるフィールドに長期フレーム・インデックスを割り当てる、3に等しいmemory\_management\_control\_operationコマンドが存在する場合、同じフレームまたは相補的参照フィールド対の他のフィールドに同じ長期フレーム・インデックスを割り当てる、別のmemory\_management\_control\_operationコマンドが、同じ復号参照ピクチャ・マーキング・シンタックス構造内に存在する。

【0052】

上記の要件は、MMC0によって参照されるフィールドが3に等しく、後で「unused for reference」としてマーキングされる場合、例えば、ピクチャ・ヘッダ内でMMC0が2に等しく、それが、フィールドが「unused for reference」としてマーキングされる原因となる場合でさえも、果たされなければならないことに留意されたい。

【0053】

相補的参照フィールド対の（復号順で）第1のフィールドが、1に等しいlong\_term\_reference\_flag、または6に等しいmemory\_management\_control\_operationコマンドを含む場合、相補的参照フィールド対の他方のフィールドのための復号参照ピクチャ・マーキング・シンタックス構造は、その他方のフィールドに同じ長期フレーム・インデックスを割り当てる、6に等しいmemory\_management\_control\_operationコマンドを含む。

【0054】

上記の要件は、相補的参照フィールド対が後で「unused for reference」としてマーキングされる場合、例えば、第2のフィールドのピクチャ・ヘッダ内でMMC0が2に等しく、それが、フィールドが「unused for reference」としてマーキングされる原因となる場合でさえも、果たされなければならないこ

10

20

30

40

50

とに留意されたい。

【0055】

`difference_of_view_id` は、現在の `memory_management_control_operation` が適用可能な `view_id` を導き出すために使用される。

【0056】

`difference_of_pic_nums` は、それ自体以外のビュー内の短期参照ピクチャに長期フレーム・インデックスを割り当てるため、またはそれ自体以外のビュー内の短期参照ピクチャを「`unused for reference`」としてマーキングするために使用される。関連する `memory_management_control_operation` が、復号プロセスによって処理される場合、`difference_of_pic_nums` から導き出された結果のピクチャ番号は、「`unused for reference`」としてマーキングされた参照ピクチャの 1 つに割り当てられたピクチャ番号であり、先に長期フレーム・インデックスに割り当てられたピクチャ番号ではない。

【0057】

結果のピクチャ番号は、以下のように制約される。

- `field_pic_flag` が 0 に等しい場合、結果のピクチャ番号は、参照フレームまたは相補的参照フィールド対に割り当てられた 1 組のピクチャ番号の 1 つである。

20

`field_pic_flag` が 0 に等しい場合、結果のピクチャ番号は、両方のフィールドが「`unused for reference`」としてマーキングされた相補的参照フィールド対、または両方のフィールドが「`unused for reference`」としてマーキングされたフレームに割り当てられたピクチャ番号でなければならないことに留意されたい。特に、`field_pic_flag` が 0 に等しい場合、単一のフィールドが「`unused for reference`」としてマーキングされた非対フィールドまたはフレームのマーキングは、1 に等しい `memory_management_control_operation` によって影響され得ない。

- それ以外の (`field_pic_flag` が 1 に等しい) 場合、結果のピクチャ番号は、参照フィールドに割り当てられた 1 組のピクチャ番号の 1 つである。

30

【0058】

`long_term_frame_idx` は、現ピクチャの `view_id` とは異なる `view_id` を有するピクチャに長期フレーム・インデックスを割り当てるために、(2 に等しい `memory_management_control_operation` とともに) 使用される。関連する `memory_management_control_operation` が、復号プロセスによって処理される場合、`long_term_frame_idx` の値は、0 以上 `Max Long Term Frame Idx` 以下の範囲内にある。

【0059】

シンタックス `difference_of_pic_nums` は、現ピクチャの `pic Num` より大きな `pic Num` を有するピクチャを選択することを可能にする。これは、マーキングをより効率的にする。

40

【0060】

表 2 に示された異なる機能の適用が以下に示される。

【0061】

`MMC0` が 1 に等しい場合、これは、短期参照ピクチャが「`unused for reference`」であるとして定義されることを表す。したがって、`pic Num X` は、

$$\text{pic Num X} = \text{Curr Pic Num} - (\text{difference_of_pic_nums})$$

50

によって指定されるとする。`viewId`は、

`viewIdX = currViewId - (difference_of_view_id)`

によって指定されるとする。

#### 【0062】

`field_pic_flag`に応じて、`picNumX`の値は、以下のように、短期参照ピクチャを「`unused for reference`」としてマーキングするために使用される。

- `field_pic_flag`が0に等しい場合、`viewIdX`によって指定されるビュー内の`picNumX`によって指定される短期参照フレームまたは短期相補的参照フィールド対、およびそのフィールドの両方が、「`unused for reference`」としてマーキングされる。10

- それ以外の(`field_pic_flag`が1に等しい)場合、`viewIdX`によって指定されるビュー内の`picNumX`によって指定される短期参照フィールドが、「`unused for reference`」としてマーキングされる。その参照フィールドが、参照フレームまたは相補的参照フィールド対の一部である場合、そのフレームまたは相補的フィールド対も、「`unused for reference`」としてマーキングされるが、他のフィールドのマーキングは変更されない。

#### 【0063】

`MMC0`が2に等しい場合、この状況は、長期参照ピクチャが「`unused for reference`」となるように変更されることを表す。`field_pic_flag`に応じて、`LongTermPicNum`の値は、以下のように、長期参照ピクチャを「`unused for reference`」としてマーキングするために使用される。20

- `field_pic_flag`が0に等しい場合、`long_term_pic_num`に等しい`LongTermPicNum`を有する長期参照フレームまたは長期相補的参照フィールド対、およびそのフィールドの両方が、「`unused for reference`」としてマーキングされる。

- それ以外の(`field_pic_flag`が1に等しい)場合、`long_term_pic_num`に等しい`LongTermPicNum`によって指定される長期参照フィールドが、「`unused for reference`」としてマーキングされる。その参照フィールドが、参照フレームまたは相補的参照フィールド対の一部である場合、そのフレームまたは相補的フィールド対も、「`unused for reference`」としてマーキングされるが、他のフィールドのマーキングは変更されない。30

#### 【0064】

`MMC0`が3に等しい場合、この状況は、`LongTermFrameIdX`を短期参照ピクチャに割り当てる(短期参照ピクチャを長期参照ピクチャにする)プロセスを表す。

#### 【0065】

シンタックス要素`difference_of_pic_nums`および`difference_of_view_id`を与えると、変数`picNumX`および`viewIdX`が、上で指定されたように取得される。`picNumX`は、`viewIdX`によって指定されるビュー内で「`used for short-term reference`」としてマーキングされ、「`non-existing`」としてマーキングされていない、フレームまたは相補的参照フィールド対もしくは非対参照フィールドを参照する。40

#### 【0066】

`long_term_frame_idX`に等しい`LongTermFrameIdX`が、長期参照フレームまたは長期相補的参照フィールド対にすでに割り当てられている場合、そのフレームまたは相補的フィールド対、およびそのフィールドの両方が、「`unused for reference`」としてマーキングされる。50

## 【0067】

LongTermFrameIdxが、非対参照フィールドにすでに割り当てられ、そのフィールドが、picNumXによって指定されるピクチャの相補的フィールドではない場合、そのフィールドは、「unused for reference」としてマーキングされる。

## 【0068】

field\_pic\_flagに応じて、LongTermFrameIdxの値は、以下のように、ピクチャを「used for short-term reference」から「used for long-term reference」にマーキングするために使用される。

- field\_pic\_flagが0に等しい場合、viewIdxによって指定されるビュー内のpicNumXによって指定される短期参照フレームまたは短期相補的参照フィールド対、およびそのフィールドの両方のマーキングが、「used for short-term reference」から「used for long-term reference」に変更され、long\_term\_frame\_idxに等しいLongTermFrameIdxが割り当てられる。

- それ以外の(field\_pic\_flagが1に等しい)場合、viewIdxによって指定されるビュー内のpicNumXによって指定される短期参照フィールドのマーキングが、「used for short-term reference」から「used for long-term reference」に変更され、long\_term\_frame\_idxに等しいLongTermFrameIdxが割り当てられる。フィールドが、参照フレームまたは相補的参照フィールド対の一部であり、同じ参照フレームまたは相補的参照フィールド対の他のフィールドも、「used for long-term reference」としてマーキングされる場合、その参照フレームまたは相補的参照フィールド対も、「used for long-term reference」としてマーキングされ、long\_term\_frame\_idxに等しいLongTermFrameIdxが割り当てられる。

## 【0069】

MMCOが4に等しい場合、そのような状況は、LongTermFrameIdx値が、max\_long\_term\_frame\_idx\_plus1 - 1に関連する値よりも大きい場合に、参照ピクチャのステータスを「used for long-term reference」から「usused for reference」に変更するために引き起こされる。

## 【0070】

変数MaxLongTermFrameIdxは、以下のように決定される。

max\_long\_term\_frame\_idx\_plus1が0に等しい場合、MaxLongTermFrameIdxは、「no long-term frame indices」に等しく設定される。

それ以外の(max\_long\_term\_frame\_idx\_plus1が0よりも大きい)場合、MaxLongTermFrameIdxは、max\_long\_term\_frame\_idx\_plus1 - 1に等しく設定される。

## 【0071】

4に等しいmemory\_management\_control\_operationコマンドは、長期参照ピクチャを「unused for reference」としてマーキングするために使用できることに留意されたい。しかし、max\_long\_term\_frame\_idx\_plus1を送信する頻度は、本明細書によって指定されず、コーダの設計者によって選択されてよい。しかし、符号器は、イントラ・リフレッシュ要求メッセージ(intra refresh request message)などのエラー・メッセージを受信したとき、4に等しいmemory\_management\_control\_operationコマンドを送信すべきである。

10

20

30

40

50

## 【0072】

5に等しいMMC0は、（上で導き出された）viewIdによって指定されるビュー内の参照ピクチャのすべてが、「unused for reference」としてマーキングされる状況を表す。すなわち、このMMC0は、特定のビューのピクチャのすべてを（各参照ピクチャを具体的に識別する必要なしに）変更する機能をコーダに提供する。このタイプの機能は、現在コーディングされているピクチャと同じビューに属するすべての参照ピクチャのステータスを変更するために呼び出すことができる。同様に、このコマンドは、現在コーディングされているピクチャに関連するビューとは同じではない特定のビューの参照ピクチャのステータスを変更するために呼び出すことができる。

## 【0073】

10

6に等しいMMC0を有することは、現ビューに関連するビュー以外のすべてのビュー内のすべての参照ピクチャが、ステータスを「unused for reference」に変更され、MaxLongTermFrameIdx変数を「no long-term frame indices」に設定される状況を表す。コマンドは、DPBを効果的に制御して、現在コーディングされているピクチャのビューに関連しないビューについての参照ピクチャのすべてを最終的に消去する。上で言及されたように、5に等しいMMC0は、特定のビューに関連する参照ピクチャのステータスを変更するためのものであるが、（6に等しい）このMMC0は、コーディングされるピクチャのビューとは同じでないビューに関連する参照ピクチャのすべてに影響する。

## 【0074】

20

7に等しいMMC0は、参照ピクチャのステータスが、「long-term reference picture」から「used for short-term reference」に変更される状況を表す。そのような参照ピクチャは、現在コーディングされているピクチャに関連するビューとは異なるビューに関連する。

## 【0075】

図7は、（シンタックス要素difference\_of\_pics\_numsを使用する代わりに）シンタックス要素difference\_of\_pics\_nums\_minus1が提示される、本発明の原理の他の実施形態を提示している。そのような変更が関与するのは、現ピクチャのpicNumXより大きいpicNumXを有するピクチャを選択できない状況に対してである。この実施形態に関連するMMC0は、（表2において）上で識別されたのと同様に作用する。

30

## 【0076】

図8は、（スライス・ヘッダなどの）ピクチャ・ヘッダのコマンドが、AVC操作においてピクチャがコーディングされている最中に、シンタックス要素コマンドslice\_header\_mvcextension()を呼び出すように修正される、本発明の原理の一代替実施形態を提示している。すなわち、この実施形態では、多視点用のMMC0コマンドは、考慮中のピクチャとは異なるビュー（aview）の参照ピクチャが考慮される先に提示されたものの代わりに、（異なるビュー（views）のすべての参照ピクチャが考慮され得る）AVC符号化の最中に、起動されることがある。

## 【0077】

40

図9は、シンタックス要素コマンドdec\_ref\_pic\_marking\_mvcextension()の構成を開示している。（図8に示されるようなピクチャ・ヘッダ/スライス・ヘッダ内で呼び出される）この新しいシンタックスは、適切なdifference\_of\_view\_idシンタックスを設定することによって、現ビュー以外のビュー内にあるピクチャをマーキングするために使用される。コーディングされるピクチャに関連するビューに関連する参照ピクチャが、そのメモリ・ステータスを変更されることを可能にするため、difference\_of\_view\_idシンタックスは0に設定される。この提案されるシンタックス要素は、コマンドdec\_ref\_pic\_markingを使用するDPB管理用の既存のAVC関数に取って代わる。dec\_ref\_pic\_marking\_mvcextension()に関連する様々なシ

50

ンタックス要素が以下で説明される。

【0078】

現在コーディングされているピクチャの参照マーキング・モードを選択するために、`mvc_adaptive_ref_pic_marking_mode_flag`が使用される。フラグは、「0」では、スライディング・ウィンドウ参照ピクチャ・マーキング・モードを表し、短期参照ピクチャが FIFO に基づいて DPB 内に割り当てられる。フラグは、「1」では、適応参照ピクチャ・マーキング・モードを表し、参照ピクチャを「unused for reference」としてマーキングし、「long-term frame indices」を割り当てるために、要素が提供されることができる。

10

【0079】

`mvc_adaptive_ref_pic_marking_mode_flag`は、現在「used for long-term reference」としてマーキングされているフレーム、相補的フィールド対、および非対フィールドの数が、`Max(Num_ref_frames, 1)`に等しい場合、1に等しくなる。

【0080】

`memory_management_control_operation`(MMCO)は、参照ピクチャ・マーキングに影響を及ぼすために適用される制御操作を規定する。MMCO シンタックス要素には、MMCO の値によって指定される操作にとって必要なデータが後続する。呼び出される MMCO に関する値および制御操作は、(以下の)表 3 に示される。本実施形態の MMCO シンタックス要素は、それらがスライス・ヘッダ内に出現した順序で、復号プロセスによって処理され、各`memory_management_control_operation`に対して述べられたセマンティック制約は、その個々の MMCO が処理される特定の位置において、その順序で適用される。

20

【0081】

`memory_management_control_operation`の解釈のため、参照ピクチャという用語は、以下のように解釈される。

- 現ピクチャがフレームである場合、参照ピクチャという用語は、参照フレームまたは相補的参照フィールド対を指示する。

- それ以外の(現ピクチャがフィールドである)場合、参照ピクチャという用語は、参照フィールドまたは参照フレームのフィールドを指示する。

30

【0082】

`memory_management_control_operation`は、`memory_management_control_operation`が復号プロセスによって処理される場合に、指定された参照ピクチャが「used for short-term reference」としてマーキングされない限り、スライス・ヘッダ内において、1に等しくならない。

【0083】

`memory_management_control_operation`は、`memory_management_control_operation`が復号プロセスによって処理される場合に、指定された長期ピクチャ番号が「used for long-term reference」としてマーキングされた参照ピクチャを指示しない限り、スライス・ヘッダ内において、2に等しくならない。

40

【0084】

`memory_management_control_operation`は、`memory_management_control_operation`が復号プロセスによって処理される場合に、指定された参照ピクチャが「used for short-term reference」としてマーキングされない限り、スライス・ヘッダ内において、3に等しくならない。

【0085】

50

`memory_management_control_operation` は、`memory_management_control_operation` が復号プロセスによって処理される場合に、変数 `Max Long Term Frame Id x` の値が「`no long-term frame indices`」に等しい場合、3、5、または6に等しくならない。

【0086】

フィールドを復号する際に、短期参照フレームの一部または短期相補的参照フィールド対の一部であるフィールドに長期フレーム・インデックスを割り当てる、3に等しい`memory_management_control_operation`コマンドが存在する場合、同じフレームまたは相補的参照フィールド対の他のフィールドに同じ長期フレーム・インデックスを割り当てる、別の`memory_management_control_operation`コマンドが、同じ復号参照ピクチャ・マーキング・シンタックス構造内に存在する。

【0087】

上記の要件は、3に等しい`memory_management_control_operation`によって参照されるフィールドが、後で「`unused for reference`」としてマーキングされる場合（例えば、2に等しい`memory_management_control_operation`が同じスライス・ヘッダ内に存在し、それが、フィールドが「`unused for reference`」としてマーキングされる原因となる場合）でさえも、果たされなければならないことに留意されたい。

【0088】

相補的参照フィールド対の（復号順で）第1のフィールドが、1に等しい`long_term_reference_flag`、または6に等しい`memory_management_control_operation`コマンドを含む場合、相補的参照フィールド対の他方のフィールドのための復号参照ピクチャ・マーキング・シンタックス構造は、その他方のフィールドに同じ長期フレーム・インデックスを割り当てる、6に等しい`memory_management_control_operation`コマンドを含む。

【0089】

上記の要件は、相補的参照フィールド対の第1のフィールドが後で「`unused for reference`」としてマーキングされる場合（例えば、2に等しい`memory_management_control_operation`が第2のフィールドのスライス・ヘッダ内に存在し、それが、第1のフィールドが「`unused for reference`」としてマーキングされる原因となる場合）でさえも、果たされなければならないことに留意されたい。

【0090】

`difference_of_view_id` は、現在の`memory_management_control_operation`が適用可能な`view_id`を導き出すために使用される。

【0091】

`difference_of_pic_nums` は、短期参照ピクチャに長期フレーム・インデックスを割り当てるため、または短期参照ピクチャを「`unused for reference`」としてマーキングするために使用される。関連する`memory_management_control_operation`が、復号プロセスによって処理される場合、`difference_of_pic_nums` から導き出された結果のピクチャ番号は、「`unused for reference`」としてマーキングされた参照ピクチャの1つに割り当てられたピクチャ番号であり、先に長期フレーム・インデックスに割り当てられたピクチャ番号ではない。結果のピクチャ番号は、以下のように制約される。

10

20

30

40

50

- `field_pic_flag` が 0 に等しい場合、結果のピクチャ番号は、参照フレームまたは相補的参照フィールド対に割り当てられた 1 組のピクチャ番号の 1 つである。`field_pic_flag` が 0 に等しい場合、結果のピクチャ番号は、両方のフィールドが「`unused for reference`」としてマーキングされた相補的参照フィールド対、または両方のフィールドが「`unused for reference`」としてマーキングされたフレームに割り当てられたピクチャ番号でなければならぬことに留意されたい。特に、`field_pic_flag` が 0 に等しい場合、単一のフィールドが「`unused for reference`」としてマーキングされた非対フィールドまたはフレームのマーキングは、1 に等しい`memory_management_control_operation`によって影響され得ない。

- それ以外の(`field_pic_flag` が 1 に等しい)場合、結果のピクチャ番号は、参照フィールドに割り当てられた 1 組のピクチャ番号の 1 つである。

#### 【0092】

`long_term_pic_num` は、長期参照ピクチャを「`unused for reference`」としてマーキングするために、(2 に等しい`memory_management_control_operation`とともに) 使用される。関連する`memory_management_control_operation` が、復号プロセスによって処理される場合、`long_term_pic_num` は、「`used for long-term reference`」として現在マーキングされている参照ピクチャの 1 つに割り当てられた長期ピクチャ番号に等しくなる。

#### 【0093】

結果の長期ピクチャ番号は、以下のように制約される。

- `field_pic_flag` が 0 に等しい場合、結果の長期ピクチャ番号は、参照フレームまたは相補的参照フィールド対に割り当てられた 1 組の長期ピクチャ番号の 1 つである。

`field_pic_flag` が 0 に等しい場合、結果の長期ピクチャ番号は、両方のフィールドが「`used for reference`」としてマーキングされた相補的参照フィールド対、または両方のフィールドが「`used for reference`」としてマーキングされたフレームに割り当てられた長期ピクチャ番号でなければならぬことに留意されたい。特に、`field_pic_flag` が 0 に等しい場合、単一のフィールドが「`used for reference`」としてマーキングされた非対フィールドまたはフレームのマーキングは、2 に等しい`memory_management_control_operation`によって影響され得ない。

- それ以外の(`field_pic_flag` が 1 に等しい)場合、結果の長期ピクチャ番号は、参照フィールドに割り当てられた 1 組の長期ピクチャ番号の 1 つである。

#### 【0094】

`long_term_frame_idx` は、ピクチャに長期フレーム・インデックスを割り当てるために、(3 または 6 に等しい`memory_management_control_operation`とともに) 使用される。関連する`memory_management_control_operation` が、復号プロセスによって処理される場合、`long_term_frame_idx` の値は、0 以上`MaxLongTermFrameIdx` 以下の範囲内にある。

#### 【0095】

シンタックス`difference_of_pic_nums` は、現ピクチャの`picNum`より大きな`picNum`を有するピクチャを選択することを可能にする。これは、マーキングをより効率的にする。

#### 【0096】

異なるMMC0コマンドに対する復号参照ピクチャ・マーキング・プロセスが、以下で説明される。`viewId` は、

`viewId = CurrentViewId - (difference_of_view`

10

20

30

40

50

— i d )

によって指定されるとする。

【 0 0 9 7 】

(以下の表 3 に示される)すべてのMMC0コマンドは、viewIdxとして上で導き出されたviewIdに適用される。

【 0 0 9 8 】

【表 3】

memory_management_control_operation	メモリ管理制御操作
0	memory_management_control_operation シンタックス要素のループを終了する
1	短期参照ピクチャを「unused for reference (参照に使用されない)」としてマーキングする
2	長期参照ピクチャを「unused for reference (参照に使用されない)」としてマーキングする
3	短期参照ピクチャを「used for long-term reference (長期参照に使用される)」としてマーキングし、それに長期フレーム・インデックスを割り当てる
4	最大長期フレーム・インデックスを指定し、最大値よりも大きな長期フレーム・インデックスを有するすべての長期参照ピクチャを「unused for reference (参照に使用されない)」としてマーキングする
5	すべての参照ピクチャを「unused for reference (参照に使用されない)」としてマーキングし、MaxLongTermFrameIdx 変数に「no long-term frame indices (長期フレーム・インデックスでない)」を設定する
6	現ピクチャを「used for long-term reference (長期参照に使用される)」としてマーキングし、それに長期フレーム・インデックスを割り当てる
7	それ自体以外のビュー内の長期参照ピクチャを「used for short-term reference (短期参照に使用される)」としてマーキングする

表 3

10

20

30

40

【 0 0 9 9 】

MMC0が0に等しい場合、ピクチャ・ヘッダ(例えば、スライス・ヘッダ)のマーキングは終了させられる。

【 0 1 0 0 】

MMC0が1に等しい場合、特定の参照フレームは、それに関連するステータスを「short-term reference picture」から「unused for reference」に変更される。picNumXは、

50

```
picNumX = CurrPicNum - (difference_of_pic_nums)
```

によって指定されるとする。

#### 【0101】

加えて、field\_pic\_flagに応じて、picNumXの値は、以下のように、短期参照ピクチャを「unused for reference」としてマーキングするために使用される。

- field\_pic\_flagが0に等しい場合、viewIdXによって指定されるビュー内のpicNumXによって指定される短期参照フレームまたは短期相補的参考フィールド対、およびそのフィールドの両方が、「unused for reference」としてマーキングされる。10

- それ以外の(field\_pic\_flagが1に等しい)場合、viewIdXによって指定されるビュー内のpicNumXによって指定される短期参照フィールドが、「unused for reference」としてマーキングされる。その参照フィールドが、参照フレームまたは相補的参照フィールド対の一部である場合、そのフレームまたは相補的フィールド対も、「unused for reference」としてマーキングされるが、他のフィールドのマーキングは変更されない。

#### 【0102】

MMC0が2に等しい場合、特定の参照ピクチャは、それに関連するステータスを「long-term reference picture」から「unused for reference」に変更される。field\_pic\_flagに応じて、LongTermPicNumの値は、以下のように、長期参照ピクチャを「unused for reference」としてマーキングするために使用される。20

- field\_pic\_flagが0に等しい場合、long-term\_pic\_numに等しいLongTermPicNumを有する長期参照フレームまたは長期相補的参照フィールド対、およびそのフィールドの両方が、「unused for reference」としてマーキングされる。

- それ以外の(field\_pic\_flagが1に等しい)場合、long-term\_pic\_numに等しいLongTermPicNumによって指定される長期参照フィールドが、「unused for reference」としてマーキングされる。その参照フィールドが、参照フレームまたは相補的参照フィールド対の一部である場合、そのフレームまたは相補的フィールド対も、「unused for reference」としてマーキングされるが、他のフィールドのマーキングは変更されない。30

#### 【0103】

MMC0が3に等しい場合、特定の参照フレームは、LongTermFrameIdxを割り当てられ、短期参照ピクチャを長期参照ピクチャに割り当てる。シンタックス要素difference\_of\_pic\_numsおよびdifference\_of\_view\_idを与えると、変数picNumXおよびviewIdXが、上で指定されたように取得される。picNumXは、viewIdXによって指定されるビュー内で「used for short-term reference」としてマーキングされ、「non-existing」としてマーキングされていない、フレームまたは相補的参照フィールド対もしくは非対参照フィールドを参照する。40

#### 【0104】

long\_term\_frame\_idxに等しいLongTermFrameIdxが、長期参照フレームまたは長期相補的参照フィールド対にすでに割り当てられている場合、そのフレームまたは相補的フィールド対、およびそのフィールドの両方が、「unused for reference」としてマーキングされる。LongTermFrameIdxが、非対参照フィールドにすでに割り当てられ、そのフィールドが、picNumXによって指定されるピクチャの相補的フィールドではない場合、そのフィールドは、「unused for reference」としてマーキングされる。field50

`d_pic_flag`に応じて、`LongTermFrameIdx`の値は、以下のように、ピクチャを「`used for short-term reference`」から「`used for long-term reference`」にマーキングするために使用される。

- `field_pic_flag`が0に等しい場合、`viewIdx`によって指定されるビュー内の`picNumX`によって指定される短期参照フレームまたは短期相補的参照フィールド対、およびそのフィールドの両方のマーキングが、「`used for short-term reference`」から「`used for long-term reference`」に変更され、`long_term_frame_idx`に等しい`LongTermFrameIdx`が割り当てられる。10

- それ以外の(`field_pic_flag`が1に等しい)場合、`viewIdx`によって指定されるビュー内の`picNumX`によって指定される短期参照フィールドのマーキングが、「`used for short-term reference`」から「`used for long-term reference`」に変更され、`long_term_frame_idx`に等しい`LongTermFrameIdx`が割り当てられる。フィールドが、参照フレームまたは相補的参照フィールド対の一部であり、同じ参照フレームまたは相補的参照フィールド対の他のフィールドも、「`used for long-term reference`」としてマーキングされる場合、その参照フレームまたは相補的参照フィールド対も、「`used for long-term reference`」としてマーキングされ、`long_term_frame_idx`に等しい`LongTermFrameIdx`が割り当てられる。20

#### 【0105】

MMC0が4に等しい場合、最大長期フレーム・インデックス値が指定され、それによって、長期参照フレームとして識別され、最大値より大きなフレーム・インデックスを有する参照フレームのすべてが、「`unused for reference`」であるとして分類される。具体的には(関数呼出しの用語法のなかで)、`LongTermFrameIdx`が`max_long_term_frame_idx_plus1 - 1`より大きく、「`used for long term reference`」としてマーキングされた、すべてのピクチャが、「`usused for reference`」としてマーキングされる。30

#### 【0106】

変数`MaxLongTermFrameIdx`は、以下のように導き出される。

- `max_long_term_frame_idx_plus1`が0に等しい場合、`MaxLongTermFrameIdx`は、「`no long-term frame indices`」に等しく設定される。

- それ以外の(`max_long_term_frame_idx_plus1`が0より大きい)場合、`MaxLongTermFrameIdx`は、`max_long_term_frame_idx_plus1 - 1`に等しく設定される。このMMC0は、長期参照ピクチャを「`unused for reference`」としてマーキングするために使用できることに留意されたい。しかし、`max_long_term_frame_idx_plus1`を送信する頻度は、本発明に準拠するコーダの設計者によって決定されるべきである。しかし、コーダは、イントラ・リフレッシュ要求メッセージなどのエラー・メッセージを受信したとき、4に等しい`memory_management_control_operation`コマンドを送信すべきである。40

#### 【0107】

MMC0が5に等しい場合、コーダが、`viewIdx`によって指定されるビュー(特定のビュー)内のすべての参照ピクチャを「`unused for reference`」としてマーキング、`MaxLongTermFrameIdx`変数を「`no long-term frame indices`」に等しく設定する。これは、特定のビューによって識別される参照ピクチャが、そのような参照ピクチャが「`long-term`」で50

あるとしてマーキングされる前に、「`unused for reference`」となるように設定されることを意味する。

【0108】

MMC0が6に等しい場合、コーディングされる現ピクチャが、「`used for long-term`」であるとしてマーキングされ、長期フレーム・インデックスが、そのピクチャに割り当てられる。具体的には、`long_term_frame_idx`に等しい変数`LongTermFrameIdx`が、長期参照フレームまたは長期相補的参照フィールド対にすでに割り当てられている場合、そのフレームまたは相補的フィールド対、およびそのフィールドの両方が、「`unused for reference`」としてマーキングされる。`LongTermFrameIdx`が、非対参照フィールドにすでに割り当てられ、そのフィールドが、現ピクチャの相補的フィールドではない場合、そのフィールドは、「`unused for reference`」としてマーキングされる。10

【0109】

現ピクチャは、「`used for long-term reference`」としてマーキングされ、`long_term_frame_idx`に等しい`LongTermFrameIdx`が割り当てられる。

【0110】

`field_pic_flag`が0に等しい場合、そのフィールドの両方も、「`used for long-term reference`」としてマーキングされ、`long_term_frame_idx`に等しい`LongTermFrameIdx`が割り当てられる。20

【0111】

`field_pic_flag`が1に等しく、現ピクチャが相補的参照フィールド対の(復号順で)第2のフィールドであり、相補的参照フィールド対の第1のフィールドも「`used for long-term reference`」として現在マーキングされている場合、その相補的参照フィールド対も、「`used for long-term reference`」としてマーキングされ、`long_term_frame_idx`に等しい`LongTermFrameIdx`が割り当てられる。

【0112】

現復号参照ピクチャをマーキングした後、少なくとも1つのフィールドが「`used for reference`」としてマーキングされたフレームの総数と、少なくとも1つのフィールドが「`used for reference`」としてマーキングされた相補的フィールド対の数と、「`used for reference`」としてマーキングされた非対フィールドの数とを加えた数は、`Max(num_ref_frames, 1)`より大きくならない。いくつかの状況では、上述の言明は、1、2、または4に等しい`memory_management_control_operation`シンタックス要素と比べて、6に等しい`memory_management_control_operation`シンタックス要素が復号参照ピクチャ・マーキング・シンタックス内において出現できる順序に、制約を課すことがあることに留意されたい。30

【0113】

MMC0が7に等しい場合、`viewIdx`によって識別されるビュー内の`long_term_pic_num`によって識別される長期参照ピクチャが、「`used for short-term reference`」としてマーキングされる。これは、(pic番号によって識別される)特定のフレームおよび指定されたビューが、そのステータスを「`long-term reference picture`」から「`short-term reference picture`」に変更されることを意味する。

【0114】

本発明の一代替実施形態が、図10に示されるシンタックス要素により開示されており、「`difference_of_pic_nums`」または「`difference_of_`50

`pic_nums_minus 1` が、 `difference_of_view_id` の値に基づいて送信される。このソリューションは、図 8 について説明された `difference_of_pic_nums` シンタックス要素を使用する代わりに、時間的ケースのためだけに MMC0 コマンドを使用する場合に提案される。

【0115】

図 11 は、本発明の原理による、一般的な参照ピクチャ・マーキング方法のブロック図 1100 を開示している。

【0116】

ステップ 1105 は、（一般にコーディング操作がグループ・オブ・ビデオ・ベースの動画からのピクチャの符号化である）ピクチャをコーディングする一般概念を表している。この操作は、コーディングされたピクチャの復号も表すことができる。 10

【0117】

しかし、このステップでは、コーディングされるピクチャは、多視点ビデオ符号化システムにおいて使用される複数のビューからの特定のビューに関連づけられる。好ましい実施形態は、AVC コーダのコンテクスト内で、MVC ビデオ規格において開示された原理を使用するが、他の多視点ビデオ規格が使用されてよいことを理解されたい。重要なこととしては、コーディングされるピクチャは、ピクチャ ID 番号およびビュー ID 番号に関連づけられる。ピクチャ ID は、コーディング・ピクチャ系列内におけるコーディングされるピクチャ数を表す。コーディングされるピクチャは、コーディングされるピクチャが関連するビューに対応する、ビュー ID 番号（1 から  $n$  まで、 $n$  はビューの総数）も有する。例えば、ビュー「2」に関連するコーディングされるピクチャは、「2」のビュー ID 番号を有するものとして知られる。 20

【0118】

ステップ 1110 において、コーディングされるピクチャは、（DPB などの）メモリ・デバイスに保存され、コーディングされるピクチャは、メモリ・ステータスを割り当てられる。コーディングされるピクチャは、そのピクチャが参照ピクチャとして使用できるように保存される。上述のように、コーディングされるピクチャは、関連する少なくとも 3 つの異なるメモリ・ステータスを有することができる。

「Long-Term Reference Picture」は、コーディングされるピクチャが参照ピクチャとして保存されるべき場合を表す。長期参照ピクチャとして指定されたコーディングされるピクチャは、（長期ピクチャ・インデックスの）インデックス番号を割り当てられる。このピクチャは、しばらくの間保持されるものとされ、その結果、そのようなピクチャは、将来のピクチャをコーディングする場合に、参照ピクチャとして使用することができる。 30

「Short-Term Reference Picture」は、参照ピクチャとして短期間保持されるべきコーディングされるピクチャを表す。この場合、参照ピクチャは、空き領域が必要とされた場合、メモリ・デバイス（DPB）から取り除かれる。

「Unused as Reference」は、コーディングされるピクチャが参照ピクチャとして使用されることを意図されていない場合を表す。この場合、DPB は、空き領域が必要とされた場合に（LIFO/FIFO を使用して）参照ピクチャを削除することができ、または DPB から直ちに消去される。 40

【0119】

`unused as reference` であるとして指定されたピクチャは、直ちに削除されてもよく、ステップ 1110 およびステップ 1115 の最中に DPB に決して保存されなくてもよいとすることも可能である。

【0120】

ステップ 1120 において、第 2 のビデオ・ピクチャがコーディングされる。この時点で、ビデオ・ピクチャは、（他の実施形態で説明されたように）メモリ管理コマンド操作を用いてマーキングされ、MMC0 は、参照ピクチャを保存するために使用される DPB などのメモリ・デバイス内に保存された参照ピクチャに影響を及ぼす。 50

## 【0121】

先に説明されたように、参照ピクチャに関連づけられたメモリ・ステータスをどのように変更するかを決定するために、異なるMMC0が利用できる。そのような変更は、現在コーディングされているピクチャが保存された参照ピクチャと同じ／異なるビューを有するかどうかに基づいて、行われることができる（例えば、同じビューIDを有するすべてのピクチャは同時に変更され、これは大域的変更を表す）。ピクチャに関連づけられたメモリ・ステータスの変更は、直接的に行われてよく、その場合、特定の保存された参照ピクチャが識別され、MMC0は、そのピクチャのメモリ・ステータスが変更されることを指定する（例えば、メモリ・ステータスを「long-term reference」、「short-term reference」および「unused as reference」の間で変更し、これは局所的変更を表す）。

## 【0122】

加えて、上述のように、本発明の様々な実施は、現在コーディングされているピクチャのビューIDが、異なるビューIDを有する（異なるビュー間にわたる）参照ピクチャのために、またはコーディングされるピクチャと同じビューIDを有する（特定のビューについての）参照ピクチャだけのために、ピクチャが変更され得るかどうかに影響を及ぼす状況を可能にする。

## 【0123】

ステップ1125は、MCC0コマンドの実施であり、参照ピクチャを保存するメモリ記憶デバイスは、その参照ピクチャに関連づけられたメモリ・ステータスを実施する。これらのタイプの操作も上述されている。

## 【0124】

本原理の上記および他の特徴および利点は、本明細書の教示に基づいて、当業者によって容易に確認できよう。本原理の教示は、ハードウェア、ソフトウェア、ファームウェア、専用プロセッサ、またはそれらの組合せといった様々な形態で実施できることを理解されたい。

## 【0125】

最も好ましくは、本原理の教示は、ハードウェアとソフトウェアの組合せとして実施される。さらに、ソフトウェアは、プログラム記憶ユニット上で有形に実施されるアプリケーション・プログラムとして実施されてよい。アプリケーション・プログラムは、任意の適切なアーキテクチャを備えるマシンによって、アップロードされ、実行されてよい。好ましくは、マシンは、1つまたは複数の中央処理装置（「CPU」）、ランダム・アクセス・メモリ（「RAM」）、および入出力（「I/O」）インターフェースなどのハードウェアを有する、コンピュータ・プラットフォーム上で実施される。コンピュータ・プラットフォームは、オペレーティング・システムおよびマイクロ命令コードも含んでよい。本明細書で説明された様々なプロセスおよび関数は、CPUによって実行され得る、マイクロ命令コードの部分もしくはアプリケーション・プログラムの部分、またはそれらの任意の組合せとすることができます。加えて、追加のデータ記憶ユニットおよび印刷ユニットなど、他の様々な周辺ユニットが、コンピュータ・プラットフォームに接続されてよい。

## 【0126】

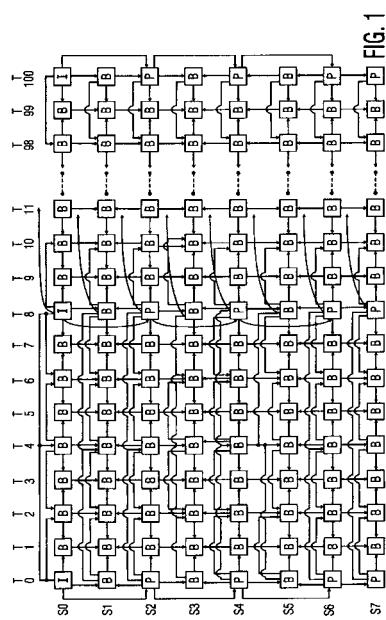
添付の図面に示される構成要素システム・コンポーネントおよび方法のいくつかは、好ましくはソフトウェアで実施されるので、システム・コンポーネントまたはプロセス機能ロックの間の実際の接続は、本原理がプログラムされる方法に応じて異なり得ることをさらに理解されたい。本明細書の教示を与えれば、当業者は、本原理の上記および類似の実施または構成を企図することができる。

## 【0127】

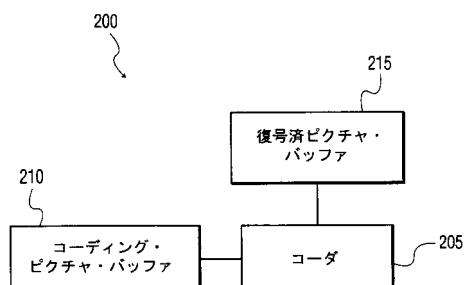
説明的な実施形態が、添付の図面を参照しながら本明細書で説明されたが、本原理は、それらの実施形態そのままで限定されず、本原理の範囲または主旨から逸脱することなく、当業者によって、それらの実施形態に様々な変更および修正が施され得ることを理解されたい。そのような変更および修正のすべては、添付の特許請求の範囲において説明され

る本原理の範囲内に含まれることが意図されている。

【図1】



【図2】



【図3】

dec_rel_pic_marking()	C	ディスクリプタ
if(nal_unit_type == 5    nal_unit_type == 21) { /* nal_unit_type 21 is specified in Annex Error! Reference source not found. */ no_output_of_prior_pics_flag	215	u(1)
long_term_reference_flag	215	u(1)
} else { adaptive_rel_pic_marking_mode_flag	215	u(1)
if(adaptive_rel_pic_marking_mode_flag)		
do { memory_management_control_operation	215	ue(v)
if(memory_management_control_operation == 1    memory_management_control_operation == 3) difference_of_pic_nums_minus1	215	ue(v)
if(memory_management_control_operation == 2)		
long_term_pic_num	215	ue(v)
if(memory_management_control_operation == 3    memory_management_control_operation == 6) long_term_frame_idx	215	ue(v)
if(memory_management_control_operation == 4)		
max_long_term_frame_idx_plus1	215	ue(v)
} while(memory_management_control_operation != 0)		
} }		

FIG. 3

【図4】

seq_parameter_set_mvc_extension()	C	ディスクリプタ
num_views_minus_1		ue(v)
for(i = 0; i < num_views_minus_1; i++) { num_multiview_refs_for_list0[i]		ue(v)
num_multiview_refs_for_list1[i]		ue(v)
for(j = 0; j < num_multiview_refs_for_list0; j++) { anchor_reference_view_for_list_0[i][j]		u(10)
non_anchor_reference_view_for_list_0[i][j]		u(10)
} for(j = 0; j < num_multiview_refs_for_list1; j++) { anchor_reference_view_for_list_1[i][j]		u(10)
non_anchor_reference_view_for_list_1[i][j]		u(10)
} }		

FIG. 4

【図5】

dec_rel_pic_marking_mvc_extension()	C	ディスクリプタ
mvc_adaptive_rel_pic_marking_mode_flag	215	u(1)
if(mvc_adaptive_rel_pic_marking_mode_flag)		
do { memory_management_control_operation	215	ue(v)
if(memory_management_control_operation == 1    memory_management_control_operation == 3    memory_management_control_operation == 7) difference_of_view_id	215	se(v)
difference_of_pic_nums	215	se(v)
} if(memory_management_control_operation == 2    memory_management_control_operation == 7) long_term_pic_num	215	ue(v)
if(memory_management_control_operation == 4) max_long_term_frame_idx_plus1	215	ue(v)
if(memory_management_control_operation == 5) difference_of_view_id_minus1	215	ue(v)
if(memory_management_control_operation == 3) long_term_frame_idx	215	ue(v)
} while(memory_management_control_operation != 0)		
dec_rel_pic_marking()		
}		

FIG. 5

【図6】

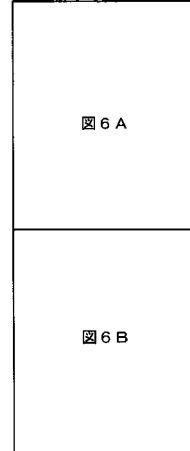


FIG. 6

【図 6 A】

slice_header_mvc_extension()	C	ディスクリプタ
first_mb_in_slice	2	ue(v)
slice_type	2	ue(v)
pic_parameter_set_id	2	ue(v)
frame_num	2	u(v)
if(frame_mbs_only_flag){		
field_pic_flag	2	u(1)
if(field_pic_flag){		
bottom_field_flag	2	u(1)
}		
if(nal_unit_type == 21){		
idr_pic_id	2	ue(v)
if(pic_order_cnt_type == 0){		
pic_order_cnt_ls1	2	u(v)
if(pic_order_present_flag & !field_pic_flag){		
delta_pic_order_cnt_bottom	2	se(v)
}		
if(pic_order_cnt_type == 1 & !delta_pic_order_always_zero_flag){		
delta_pic_order_cnt[0]	2	se(v)
if(pic_order_present_flag & !field_pic_flag){		
delta_pic_order_cnt[1]	2	se(v)
}		
if(redundant_pic_cnt_present_flag){		
redundant_pic_cnt	2	ue(v)
if(slice_type == EB){		
direct_spatial_mv_pred_flag	2	u(1)
if(slice_type == EB    slice_type == EB){		
num_ref_idx_active_override_flag	2	u(1)
if(num_ref_idx_active_override_flag){		
num_ref_idx_10_active_minus1	2	ue(v)
if(slice_type == EB){		
num_ref_idx_11_active_minus1	2	ue(v)
}		
}		

FIG. 6A

【図 6 B】

ref_pic_list_reordering()	2	
if( ( weighted_pred_flag & ( slice_type == EP ) )    ( weighted_bipred_idc == 1 & slice_type == EB ) )		
pred_weight_table[ ]	2	
if(nal_ref_idc != 0)		
dec_ref_pic_marking_mvc_extension()	2	
if(entropy_coding_mode_flag & slice_type != 1 & slice_type != SI)		
cabac_init_idc	2	ue(v)
slice_qp_delta	2	se(v)
if(slice_type == SP    slice_type == SI){		
if(slice_type == SP){		
sp_for_switch_flag	2	u(1)
slice_qs_delta	2	se(v)
}		
if(deblocking_filter_control_present_flag){		
disable_deblocking_filter_idc	2	ue(v)
if(disable_deblocking_filter_idc != 1){		
slice_alpha_c0_offset_div2	2	se(v)
slice_beta_offset_div2	2	se(v)
}		
}		
if( num_slice_groups_minus1 > 0 && slice_group_map_type >= 3 && slice_group_map_type <= 5 )		
slice_group_change_cycle	2	u(v)
}		

FIG. 6B

【図 7】

dec_ref_pic_marking_mvc_extension()	C	ディスクリプタ
mvc_adaptive_ref_pic_marking_mode_flag	215	u(1)
if(mvc_adaptive_ref_pic_marking_mode_flag){		
do{		
memory_management_control_operation	215	ue(v)
if(memory_management_control_operation == 1    memory_management_control_operation == 3){		
difference_of_view_id	215	se(v)
difference_of_pic_nums_minus1	215	se(v)
}		
if(memory_management_control_operation == 2){		
long_term_pic_num	215	ue(v)
if(memory_management_control_operation == 4){		
max_long_term_frame_idx_plus1	215	ue(v)
if(memory_management_control_operation == 5){		
difference_of_view_id	215	se(v)
if(memory_management_control_operation == 3){		
long_term_frame_idx	215	ue(v)
}		
dec_ref_pic_marking()		

FIG. 7

【図 8】

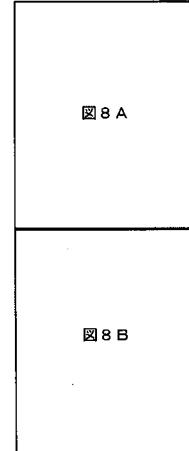


FIG. 8

【図 8 A】

slice_header_mvc_extension()		C	ディスクリプタ
first_mb_in_slice	2	ue(v)	
slice_type	2	ue(v)	
pic_parameter_set_id	2	ue(v)	
frame_num	2	u(v)	
if(frame_mbs_only_flag)			
field_pic_flag	2	u(1)	
if(field_pic_flag)			
bottom_field_flag	2	u(1)	
)			
if(nal_unit_type == 21)			
idr_pic_id	2	ue(v)	
if(pic_order_cnt_type == 0)			
pic_order_cnt_sb	2	u(v)	
if(pic_order_present_flag && !field_pic_flag)			
delta_pic_order_cnt_bottom	2	se(v)	
)			
if(pic_order_cnt_type == 1 && !delta_pic_order_always_zero_flag)			
delta_pic_order_cnt[0]	2	se(v)	
if(pic_order_present_flag && !field_pic_flag)			
delta_pic_order_cnt[1]	2	se(v)	
)			
if(redundant_pic_cnt_present_flag)			
redundant_pic_cnt	2	ue(v)	
if(slice_type == EB)			
direct_spatial_mv_pred_flag	2	u(1)	
if(slice_type == EP    slice_type == EB)			
num_ref_idx_active_override_flag	2	u(1)	
if(num_ref_idx_active_override_flag)			
num_ref_idx_10_active_minus1	2	ue(v)	
if(slice_type == EB)			
num_ref_idx_11_active_minus1	2	ue(v)	
)			
)			

FIG. 8A

【図 9】

dec_ref_pic_marking_mvc_extension()		C	ディスクリプタ
if(nal_unit_type == 5    nal_unit_type == 21)			
no_output_of_prior_pics_flag	215	u(1)	
long_term_reference_flag	215	u(1)	
else			
mvc_adaptive_ref_pic_marking_mode_flag	215	u(1)	
if(mvc_adaptive_ref_pic_marking_mode_flag)			
do			
if(memory_management_control_operation != 0    memory_management_control_operation != 6)			
difference_of_view_id	215	se(v)	
memory_management_control_operation	215	ue(v)	
if(memory_management_control_operation == 1    memory_management_control_operation == 3)			
difference_of_pic_nums	215	se(v)	
if(memory_management_control_operation == 2    memory_management_control_operation == 7)			
long_term_pic_num	215	ue(v)	
if(memory_management_control_operation == 3    memory_management_control_operation == 6)			
long_term_frame_idx	215	ue(v)	
if(memory_management_control_operation == 4)			
max_long_term_frame_idx_plus1	215	ue(v)	
} while(memory_management_control_operation != 0)			
)			

FIG. 9

【図 8 B】

ref_pic_list_reordering()	2	
if((weighted_pred_flag && (slice_type == EP))    (weighted_bipred_idc == 1 && slice_type == EB))		
pred_weight_table()	2	
if(nal_ref_idc != 0)		
dec_ref_pic_marking_mvc_extension()	2	
if(entropy_coding_mode_flag && slice_type != 1 && slice_type != SI)		
cabac_init_idc	2	ue(v)
slice_qp_delta	2	se(v)
if(slice_type == SP    slice_type == SI)		
if(slice_type == SP)		
sp_for_switch_flag	2	u(1)
slice_qs_delta	2	se(v)
}		
if(deblocking_filter_control_present_flag)		
disable_deblocking_filter_idc	2	ue(v)
if(disable_deblocking_filter_idc != 1)		
slice_alpha_c0_offset_div2	2	se(v)
slice_beta_offset_div2	2	se(v)
}		
if(num_slice_groups_minus1 > 0 && slice_group_map_type > -3 && slice_group_map_type <= 5)		
slice_group_change_cycle	2	u(v)
}		

FIG. 8B

【図 10】

dec_ref_pic_marking_mvc_extension()		C	ディスクリプタ
mvc_adaptive_ref_pic_marking_mode_flag	215	u(1)	
if(mvc_adaptive_ref_pic_marking_mode_flag)			
do			
memory_management_control_operation	215	ue(v)	
if(memory_management_control_operation == 1    memory_management_control_operation == 3)			
difference_of_view_id	215	se(v)	
difference_of_pic_nums_minus1	215	se(v)	
}			
if(memory_management_control_operation == 2)			
long_term_pic_num	215	ue(v)	
if(memory_management_control_operation == 4)			
max_long_term_frame_idx_plus1	215	ue(v)	
if(memory_management_control_operation == 5)			
difference_of_view_id	215	se(v)	
if(memory_management_control_operation == 3)			
long_term_frame_idx	215	ue(v)	
} while(memory_management_control_operation != 0)			
dec_ref_pic_marking()			
}			

FIG. 10

【図11】

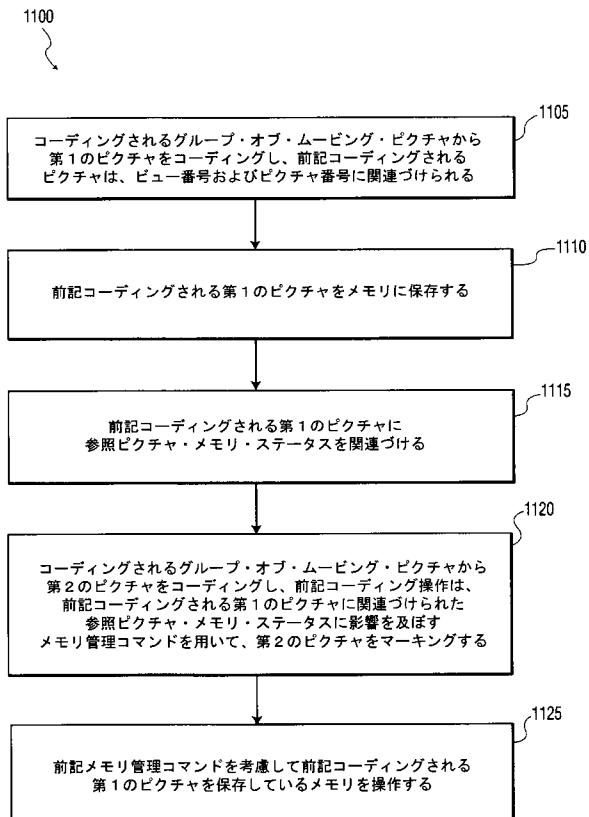


FIG. 11

---

フロントページの続き

- (72)発明者 パンディット, パービン ビブハス  
アメリカ合衆国 ニュージャージー州 フランクリン・パーク ペア・ツリー・レーン 23
- (72)発明者 スー, イエビン  
アメリカ合衆国 ワシントン州 バンクーバー ノース・イースト 第109 アベニュー アパートメント ビー8 3508
- (72)発明者 イン, ペン  
アメリカ合衆国 ニュージャージー州 ウエスト・ワインザー ワーウィック・ロード 65

審査官 岩井 健二

- (56)参考文献 特開2008-182669 (JP, A)  
国際公開第2007/114612 (WO, A1)  
国際公開第2007/081178 (WO, A1)  
国際公開第2006/001653 (WO, A1)  
Anthony Vetro et al., Joint Multiview Video Model (JMVM) 1.0, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), 20th Meeting: Klagenfurt, Austria, 2006年 8月, JVT-T208, pp.1-23  
Purvin Pandit et al., On MVC High-Level Syntax for Picture Management, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), 20th Meeting: Klagenfurt, Austria, 2006年 7月, JVT-T131, pp.1-3

- (58)調査した分野(Int.Cl., DB名)

H04N 7/24 - 7/68