

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2019-201435
(P2019-201435A)

(43) 公開日 令和1年11月21日(2019.11.21)

(51) Int.Cl.	F I	テーマコード (参考)
HO4N 19/597 (2014.01)	HO4N 19/597	5C159
HO4N 19/70 (2014.01)	HO4N 19/70	

審査請求 有 請求項の数 5 O L 外国語出願 (全 49 頁)

(21) 出願番号	特願2019-159424 (P2019-159424)	(71) 出願人	510185767 ドルビー・インターナショナル・アーベー オランダ王国, セーエン アムステルダム ズイドースト 1101, ヘリケルベル グウェグ 1-35, アポロ ビルディン グ 3エー
(22) 出願日	令和1年9月2日 (2019.9.2)	(74) 代理人	100107766 弁理士 伊東 忠重
(62) 分割の表示	特願2017-93497 (P2017-93497) の分割	(74) 代理人	100070150 弁理士 伊東 忠彦
原出願日	平成20年4月11日 (2008.4.11)	(74) 代理人	100091214 弁理士 大貫 進介
(31) 優先権主張番号	60/923,014		
(32) 優先日	平成19年4月12日 (2007.4.12)		
(33) 優先権主張国・地域又は機関	米国 (US)		
(31) 優先権主張番号	60/925,400		
(32) 優先日	平成19年4月20日 (2007.4.20)		
(33) 優先権主張国・地域又は機関	米国 (US)		

最終頁に続く

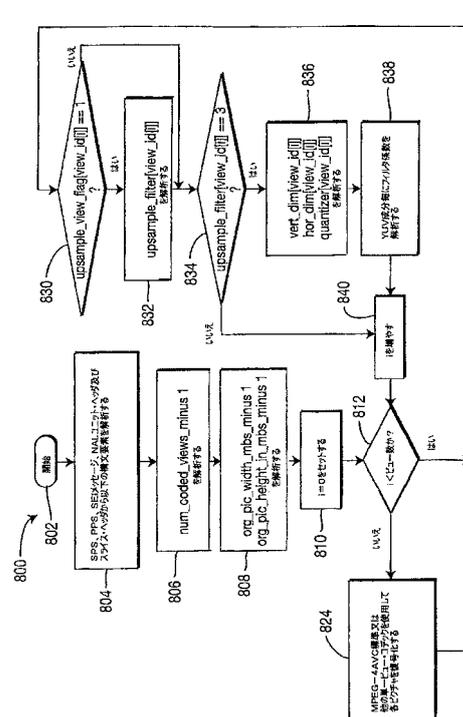
(54) 【発明の名称】 ビデオ符号化及び復号化におけるタイリング

(57) 【要約】

【課題】ビデオ符号化及び復号化におけるビューのタイリングに関する実現形態を提供する。

【解決手段】単一のピクチャに合成された複数のピクチャを含むビデオ・ピクチャにアクセスする工程と、アクセスされたビデオ・ピクチャにおける複数のピクチャがどのようにして合成されるかを示す情報にアクセスする工程(806、808)と、複数のピクチャのうちの少なくとも1つの復号化された表現を供給するようビデオ・ピクチャを復号化する工程(824)と、アクセスされた情報及び復号化されたビデオ・ピクチャを出力として供給する工程とを用いて復号する。符号化側では、単一のビデオ・ピクチャに含まれる複数のピクチャが単一のビデオ・ピクチャにどのようにして合成されるかを示す情報をフォーマット又は処理し、合成された複数のピクチャの符号化された表現をフォーマット又は処理する。

【選択図】図8A



【特許請求の範囲】

【請求項 1】

符号化されたビデオ・ピクチャを復号する装置であって、当該装置は：

符号化されたビデオ・ピクチャを含む符号化されたビデオ・ビットストリームを受領する入力部と；

プロセッサとを有しており、前記プロセッサは：

受領されたビットストリームから単一のピクチャを決定する段階であって、前記単一のピクチャは、マルチビュー・ビデオの第一のビューの第一のピクチャおよび前記マルチビュー・ビデオの第二のビューの第二のピクチャを含む、段階と；

前記第一のピクチャおよび前記第二のピクチャがどのように一緒にタイリングされているかを決定する情報にアクセスする段階であって、アクセスされる情報は、前記第一のピクチャが前記第二のピクチャの上に配置されており、前記第一のピクチャおよび前記第二のピクチャのうちの少なくとも一方が垂直方向に個別に反転されていることを示す、段階と；

前記単一のピクチャを前記第一のピクチャおよび前記第二のピクチャに復号する段階とを実行する、装置。

【請求項 2】

前記第一のピクチャおよび前記第二のピクチャの両方が垂直方向に個別に反転されている、請求項 1 記載の装置。

【請求項 3】

前記第一のピクチャのみが反転されている、請求項 1 記載の装置。

【請求項 4】

ビデオ・ピクチャを符号化されたビットストリームに符号化する装置であって、当該装置は：

マルチビュー・ビデオのビデオ・ピクチャを受領する入力部と；

プロセッサとを有しており、前記プロセッサは：

複数のピクチャを単一のビデオ・ピクチャに組み合わせる段階であって、前記複数のピクチャは前記マルチビュー・ビデオの第一のビューの第一のピクチャおよび前記マルチビュー・ビデオの第二のビューの第二のピクチャを含む、段階と；

前記複数のピクチャがどのように組み合わせられているかを示すシグナリング部を生成する段階であって、生成される情報は、前記第一のピクチャが前記第二のピクチャの上に配置されていることおよび前記第一のピクチャおよび前記第二のピクチャのうちの少なくとも一方が垂直方向に個別に反転されていることを示す、段階と；

前記単一のビデオ・ピクチャを符号化して、前記組み合わせられた複数のピクチャの符号化された表現を提供する段階と；

前記シグナリング部および前記符号化された表現を、出力として組み合わせる段階とを実行する、装置。

【請求項 5】

前記シグナリング部が、前記第一のピクチャおよび前記第二のピクチャの両方が垂直方向に反転されていることを示す、請求項 4 記載の装置。

【発明の詳細な説明】

【技術分野】

【0001】

本出願は、(1)「Multiview Information」と題する西暦 2007 年 4 月 12 日付け出願の米国特許仮出願第 60/923014 号(代理人整理番号 PU070078 号)及び(2)「View Tiling in MVC Coding」と題する西暦 2007 年 4 月 20 日付け出願の米国特許仮出願第 60/925400 号(代理人整理番号 PU0700103 号)それぞれの利益を主張する。前述の 2 つの出願

10

20

30

40

50

はそれぞれ、内容全体を本明細書及び特許請求の範囲に援用する。

【0002】

本願の原理は一般に、ビデオ符号化及び/又は復号化に関する。

【背景技術】

【0003】

ビデオ・ディスプレイ製造業者は、単一のフレーム上に別々のビューを配置又はタイリングするフレームワークを使用し得る。その場合、ビューは、そのそれぞれの位置から抽出し、レンダリングすることができる。

【発明の概要】

【発明が解決しようとする課題】

【0004】

一般的な局面によれば、単一のピクチャに合成された複数のピクチャを含むビデオ・ピクチャがアクセスされる。アクセスされたビデオ・ピクチャにおける複数のピクチャがどのようにして合成されるかを示す情報がアクセスされる。合成された複数のピクチャの復号化された表現を提供するためにビデオ・ピクチャが復号化される。アクセスされた情報及び復号化されたビデオ・ピクチャが出力として供給される。

【課題を解決するための手段】

【0005】

別の一般的な局面によれば、ビデオ・ピクチャに含まれる複数のピクチャがどのようにして単一のピクチャに合成されるかを示す情報が生成される。合成された複数のピクチャの符号化された表現を供給するためにビデオ・ピクチャが符号化される。生成された情報及び符号化されたビデオ・ピクチャが出力として供給される。

【0006】

別の一般的な局面によれば、信号又は信号構造は、単一のビデオ・ピクチャに含まれる複数のピクチャが単一のビデオ・ピクチャにどのようにして合成されるかを示す情報を含む。信号又は信号構造は、合成された複数のピクチャの符号化された表現も含む。

【0007】

別の一般的な局面によれば、単一のピクチャに合成された複数のピクチャを含むビデオ・ピクチャがアクセスされる。アクセスされたビデオ・ピクチャにおける複数のピクチャがどのようにして合成されるかを示す情報がアクセスされる。複数のピクチャのうち少なくとも1つの復号化された表現を供給するためにビデオ・ピクチャが復号化される。アクセスされた情報及び復号化された表現が出力として供給される。

【0008】

別の一般的な局面によれば、単一のピクチャに合成された複数のピクチャを含むビデオ・ピクチャがアクセスされる。アクセスされたビデオ・ピクチャにおける複数のピクチャがどのようにして合成されるかを示す情報がアクセスされる。合成された複数のピクチャの復号化された表現を供給するためにビデオ・ピクチャが復号化される。表示するために複数のピクチャの少なくとも1つを選択するユーザ入力を受け取られる。選択された少なくとも1つのピクチャの復号化された出力が供給され、復号化された出力は、アクセスされた情報、復号化された表現、及びユーザ入力に基づいて供給される。

【図面の簡単な説明】

【0009】

【図1】単一のフレーム上にタイリングされた4つのビューの例を示す図である。

【図2】単一のフレーム上で反転させ、タイリングされた4つのビューの例を示す図である。

【図3】本願の原理の実施例による、本願の原理を施すことができるビデオ符号化器を示すブロック図である。

【図4】本願の原理の実施例による、本願の原理を施すことができるビデオ復号化器を示すブロック図である。

【図5A】本願の原理の実施例による、MPEG-4AVC標準を使用して複数のビュー

10

20

30

40

50

のピクチャを符号化する方法のフロー図である。

【図 5 B】本願の原理の実施例による、MPEG-4 AVC 標準を使用して複数のビューのピクチャを符号化する方法のフロー図である。

【図 6 A】本願の原理の実施例による、MPEG-4 AVC 標準を使用して複数のビューのピクチャを復号化する方法のフロー図である。

【図 6 B】本願の原理の実施例による、MPEG-4 AVC 標準を使用して複数のビューのピクチャを復号化する方法のフロー図である。

【図 7 A】本願の原理の実施例による、MPEG-4 AVC 標準を使用して複数のビュー及び深度のピクチャを符号化する方法のフロー図である。

【図 7 B】本願の原理の実施例による、MPEG-4 AVC 標準を使用して複数のビュー及び深度のピクチャを符号化する方法のフロー図である。

【図 8 A】本願の原理の実施例による、MPEG-4 AVC 標準を使用して複数のビュー及び深度のピクチャを復号化する方法のフロー図である。

【図 8 B】本願の原理の実施例による、MPEG-4 AVC 標準を使用して複数のビュー及び深度のピクチャを復号化する方法のフロー図である。

【図 9】本願の原理の実施例による、深度信号の例を示す図である。

【図 10】本願の原理の実施例による、タイルとして加えられた深度信号の例を示す図である。

【図 11】本願の原理の実施例による、単一のフレーム上にタイリングされた 5 つのビューの例を示す図である。

【図 12】本願の原理の実施例による、本願の原理を施すことができる例示的なマルチビュー・ビデオ符号化 (MVC) 符号化器のブロック図である。

【図 13】本願の原理の実施例による、本願の原理を施すことができる例示的なマルチビュー・ビデオ符号化 (MVC) 復号化器のブロック図である。

【図 14】本願の原理の実施例による、MPEG-4 AVC 標準のマルチビュー・ビデオ符号化 (MVC) の拡張を使用したピクチャの符号化に備えて複数のビューのピクチャを処理する方法のフロー図である。

【図 15 A】本願の原理の実施例による、MPEG-4 AVC 標準のマルチビュー・ビデオ符号化 (MVC) の拡張を使用して複数のビューのピクチャを符号化する方法のフロー図である。

【図 15 B】本願の原理の実施例による、MPEG-4 AVC 標準のマルチビュー・ビデオ符号化 (MVC) の拡張を使用して複数のビューのピクチャを符号化する方法のフロー図である。

【図 16】本願の原理の実施例による、MPEG-4 AVC 標準のマルチビュー・ビデオ符号化 (MVC) の拡張を使用したピクチャの復号化に備えて複数のビューのピクチャを処理する方法のフロー図である。

【図 17 A】本願の原理の実施例による、MPEG-4 AVC 標準のマルチビュー・ビデオ符号化 (MVC) 拡張を使用して複数のビューのピクチャを復号化する方法のフロー図である。

【図 17 B】本願の原理の実施例による、MPEG-4 AVC 標準のマルチビュー・ビデオ符号化 (MVC) 拡張を使用して複数のビューのピクチャを復号化する方法のフロー図である。

【図 18】本願の原理の実施例による、MPEG-4 AVC 標準のマルチビュー・ビデオ符号化 (MVC) の拡張を使用したピクチャの符号化に備えて複数のビュー及び深度のピクチャを処理する方法のフロー図である。

【図 19 A】本願の原理の実施例による、MPEG-4 AVC 標準のマルチビュー・ビデオ符号化 (MVC) の拡張を使用して複数のビュー及び深度のピクチャを処理する方法のフロー図である。

【図 19 B】本願の原理の実施例による、MPEG-4 AVC 標準のマルチビュー・ビデオ符号化 (MVC) の拡張を使用して複数のビュー及び深度のピクチャを処理する方法の

10

20

30

40

50

フロー図である。

【図20】本願の原理の実施例による、MPEG-4 AVC標準のマルチビュー・ビデオ符号化(MVC)拡張を使用したピクチャの復号化に備えて複数のビュー及び深度のピクチャを処理する方法のフロー図である。

【図21A】本願の原理の実施例による、MPEG-4 AVC標準のマルチビュー・ビデオ符号化(MVC)の拡張を使用して複数のビュー及び深度のピクチャを復号化する方法のフロー図である。

【図21B】本願の原理の実施例による、MPEG-4 AVC標準のマルチビュー・ビデオ符号化(MVC)の拡張を使用して複数のビュー及び深度のピクチャを復号化する方法のフロー図である。

【図22】本願の原理の実施例による、画素レベルにおけるタイリング例を示す図である。

【図23】本願の原理の実施例による、本願の原理を施すことができるビデオ処理装置を示すブロック図である。

【発明を実施するための形態】

【0010】

1つ又は複数の実現形態の詳細は、添付図面及び以下明細書に記載する。特定の1つのやり方で記載されていても、実現形態は種々のやり方で構成又は実施することができる。例えば、実現形態は、方法として行うか、あるいは、動作の組を行うよう構成された装置として実施するか、あるいは、動作の組を行うための命令を記憶する装置として実施するか、あるいは、信号において実施することができる。他の局面及び特徴は、添付図面及び特許請求の範囲に関して検討される以下の詳細な説明から明らかになるであろう。

【実施例】

【0011】

種々の実現形態は、ビデオ符号化及び復号化においてビューをタイリングする方法及び装置に関する。よって、当業者は、本明細書及び特許請求の範囲に明示的に説明するか、又は示していないが、本願の原理を実施し、その趣旨及び範囲の範囲内に含まれる種々の構成を考え出すことができるであろう。

【0012】

本明細書及び特許請求の範囲記載の例及び条件付文言は全て、本願の原理、及び当該技術分野を進展させるために本願の発明者が貢献する概念の、読者の理解を支援するための教示の目的を意図しており、前述の、特記した例及び条件への限定なしであると解釈するものとする。

【0013】

更に、本願の原理、局面、及び実施例、並びにそれらの具体例を記載した、本明細書及び特許請求の範囲の記載は全て、その構造的均等物及び機能的均等物を包含することを意図している。更に、前述の均等物は、現在知られている均等物、及び将来に開発される均等物(すなわち、構造にかかわらず、同じ機能を行う、開発された何れかの構成要素)をともに含むことが意図されている。

【0014】

よって、例えば、本明細書及び特許請求の範囲に提示されたブロック図が、本願の原理を実施する例証的な回路の概念図を表すことは当業者によって認識されるであろう。同様に、フローチャート、フロー図、状態遷移図、擬似コード等は何れも、コンピュータ読み取り可能な媒体において実質的に表し、コンピュータ又はプロセッサにより、前述のコンピュータ又はプロセッサが明記されているか否かにかかわらず、実行し得る種々の処理を表す。

【0015】

図に示す種々の構成要素の機能は、専用ハードウェア、及び適切なソフトウェアに関連してソフトウェアを実行することができるハードウェアの使用によっ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提供

10

20

30

40

50

されるか、単一の共有プロセッサによって提供されるか、又は、複数の個々のプロセッサ（この一部は共有であり得る）によって提供され得る。更に、「プロセッサ」又は「コントローラ」の語を明示的に使用していることは、ソフトウェアを実行することができるハードウェアを専ら表すものと解するべきでなく、暗黙的には、限定列挙でないが、デジタル信号プロセッサ（「DSP」）ハードウェア、ソフトウェアを記憶するための読み取り専用メモリ（「ROM」）、ランダム・アクセス・メモリ（「RAM」）及び不揮発性記憶装置を含み得る。

【0016】

他のハードウェア（汎用及び/又はカスタム）も含まれ得る。同様に、図に示すスイッチは何れも概念のみである。前述の機能は、プログラム・ロジックの動作によるか、専用ロジックによるか、プログラム制御及び専用ロジックの相互作用によるか、又は手作業によって行うことができ、特定の手法は、前後関係からより具体的に分かるように実現者によって選択可能である。

10

【0017】

本願の特許請求の範囲では、特定の機能を行う手段として表される構成要素は何れも、その機能を行う何れの手段（例えば、a）その機能を行う回路構成要素の組合せや、b）機能を行うためにそのソフトウェアを実行する適切な回路と組み合わせた、ファームウェア、マイクロコード等を含む、何れかの形態のソフトウェア）も包含することが意図される。前述の特許請求の範囲で規定された本願の原理は、記載された種々の手段によって提供される機能が、請求項が要求するやり方で組み合わせられ、集約されるということに存在する。よって、前述の機能を提供することが可能な手段は何れも、本願の明細書及び特許請求の範囲記載のものと均等であるとみなされる。

20

【0018】

本願明細書における、本願の原理の「one embodiment」（若しくは「one implementation」）又は「an embodiment」（若しくは「an implementation」）への言及は、本願の実施例に関して説明した特定の構成、構造、特性等が本願の原理の少なくとも一実施例に含まれていることを意味している。よって、本明細書全体の種々の箇所に記載された「in one embodiment」又は「in an embodiment」の句は、必ずしも、同じ実施例を全て表している訳でない。

30

【0019】

「及び/又は」の語や「少なくとも一方」の語を使用していることは、例えば、「A及び/又はB」や「A及びBのうち少なくとも一方」の場合には、最初に挙げられた選択肢（A）のみの選択、第2に挙げられた選択肢（B）のみの選択、又は両方の選択肢（A及びB）の選択を包含することが意図されている。更なる例として、「A、B、及び/又はC」や「A、B及びCのうち少なくとも1つ」の場合、前述の句は、第1に挙げられた選択肢（A）のみの選択、第2に挙げられた選択肢（B）のみの選択、第3に挙げられた選択肢（C）のみの選択、第1に挙げられた選択肢及び第2に挙げられた選択肢（A及びB）のみの選択、第1に挙げられた選択肢及び第3に挙げられた選択肢（A及びC）のみの選択、第2に挙げられた選択肢及び第3に挙げられた選択肢（B及びC）のみの選択、又は、3つの選択肢（A、B及びC）全ての選択を包含することを意図している。当該技術分野及び関連技術分野において通常の知識を有する者が容易に分かるように、このことは、挙げられたいくつもの項目について拡張することができる。

40

【0020】

更に、本願の原理の1つ又は複数の実施例は本明細書及び特許請求の範囲においてMPEG-4 AVC標準に関して説明しているが、本願の原理は単にこの標準に限定されず、よって、本願の原理の趣旨を維持しながら、他の標準、勧告、及びその拡張（特に、MPEG-4 AVC標準の拡張を含むビデオ符号化標準、勧告、及びその拡張）に関して利用することができる。

【0021】

50

更に、本願の原理の他の1つ又は複数の実施例は本明細書及び特許請求の範囲において M P E G - 4 A V C 標準のマルチビュー・ビデオ符号化の拡張に関して説明しているが、本願の原理は、単にこの拡張及び/又はこの標準に限定されず、よって、本願の原理の趣旨を維持しながら、マルチビュー・ビデオ符号化に関する他のビデオ符号化の標準、勧告、及びそれらの拡張(特に、ビデオ符号化の標準、勧告、及びそれらの拡張(M P E G - 4 A V C 標準の拡張に利用することができる。マルチビュー・ビデオ符号化(M V C)は、マルチビュー系列の符号化の圧縮フレームワークである。マルチビュー・ビデオ符号化(M V C)系列は、別々の視点からの同じシーンを捕捉する2つ以上のビデオ系列の組である。

【0022】

10

更に、本明細書及び特許請求の範囲においては、ビデオ・コンテンツに関して深度情報を使用する本願の原理の他の1つ又は複数の実施例を記載しているが、本願の原理は前述の実施例に限定されず、よって、本願の原理の趣旨を維持しながら、深度情報を使用しない他の実施例を実現することができる。

【0023】

更に、本明細書及び特許請求の範囲では、「高位構文」は、階層的にマクロブロック層の上の、ビットストリーム内に存在している構文を表す。例えば、本明細書及び特許請求の範囲記載の高位構文は、スライス・ヘッダ・レベル、付加拡張情報(S E I)レベル、ピクチャ・パラメータ・セット(P P S)レベル、シーケンス・パラメータ・セット(S P S)レベル、ビュー・パラメータ・セット(V P S)レベル、及びネットワーク抽象化層(N A L)ユニット・ヘッダ・レベルの構文を表し得るが、それらに限定されない。

20

【0024】

国際標準化機構/国際電気標準会議(I S O / I E C)動画像専門家グループ-4(M P E G - 4)パート10高度ビデオ符号化(A V C)標準/国際電気通信連合(I T U - T)H.264勧告(以下、「M P E G - 4 A V C 標準」)に基づいたマルチビデオ符号化(M V C)の現在の実現形態では、参照ソフトウェアが、各ビューを単一の符号化器で符号化し、ビュー間参照を考慮に入れることにより、マルチビュー予測を達成する。各ビューは、別個のビットストリームとして符号化器により、その元の分解能で符号化され、その後、全てのビットストリームを合成して単一のビットストリームを形成する。上記単一のビットストリームはその後、復号化される。各ビューは、別個のY U V復号化出力を生成する。

30

【0025】

マルチビュー予測の別の手法は、ビューの組を疑似ビューにグループ化することを伴う。この手法の一例では、実施可能なダウンサンプリング又は他の処理により、より大きなフレーム上又はスーパー・フレーム上の合計M個のビュー(同時にサンプリングされる)のうちN個のビューからのピクチャをタイリングすることが可能である。図1に移れば、単一のフレーム上にタイリングされた4つのビューの例の全体を参照符号100で示す。4つのビューは全て、その通常の向きにある。

【0026】

図2に移れば、単一のフレーム上で反転させ、タイリングされた4つのビューの例の全体を参照符号200で示す。左上のビューは、その通常の向きにある。右上のビューは水平方向に反転させている。左下のビューは垂直方向に反転させている。右下のビューは、水平方向及び垂直方向に反転させている。よって、4つのビューが存在している場合、各ビューからのピクチャが、タイルのようなスーパー・フレームに配置される。これは、高分解能の単一の非符号化入力系列をもたらす。

40

【0027】

あるいは、画像をダウンサンプリングして、より低い分解能をもたらすことが可能である。よって、併せてタイリングされた別々のビューをそれぞれが含む複数の系列を作成する。その場合、前述の系列はそれぞれ、別々のN個のタイリングされたビューをそれぞれが含む疑似ビューを形成する。図1は一疑似ビューを示し、図2は別の疑似ビューを示す

50

。これらの疑似ビューは次いで、I S O / I E C M P E G - 2 標準やM P E G - 4 A V C 標準などの既存のビデオ符号化標準を使用して符号化することが可能である。

【0028】

マルチビュー予測の更に別の手法は、新たな標準を使用して、別々のビューを別個に符号化し、復号化後、プレイヤーによって要求されるようにビューをタイリングすることを伴う。

【0029】

更に、別の手法では、ビューは、画素単位でタイリングすることも可能である。例えば、4つのビューを含むスーパービューでは画素(x, y)はビュー0からであり得る一方、画素(x + 1, y)はビュー1からであり得、画素(x, y + 1)はビュー2からであり得、画素(x + 1, y + 1)はビュー3からであり得る。

10

【0030】

多くのディスプレイ製造業者は、単一のフレーム上に別々のビューを配置又はタイリングし、それらのそれぞれの位置からのビューを抽出し、レンダリングする前述のフレームワークを使用する。前述の場合、ビットストリームが前述の特性を有するかを判定する標準的なやり方は存在しない。よって、システムが、大きなフレームにおいて別々のビューのピクチャをタイリングする方法を使用する場合、別々のビューを抽出する方法は、そのシステム固有のものである。

【0031】

しかし、ビットストリームが前述の特性を有するかを判定する標準的なやり方は存在しない。本願の発明者は、表示や他の後処理を支援するためにレンダリング装置又はプレイヤーが前述の情報を抽出することを容易にするために高位構文を提案している。サブピクチャが、異なる分解能を有している可能性もあり、最終的にビューをレンダリングするために多少のアップサンプリングが必要であり得る。ユーザは、アップサンプルを高位の構文でも示させたいことがあり得る。更に、深度フォーカスを変えるためのパラメータを送信することも可能である。

20

【0032】

一実施例では、別のビューに属するサブピクチャを各ピクチャが含むM P E G - 4 A V C 標準準拠ビットストリームにおけるマルチビュー情報のシグナリングのための新たな付加拡張情報(S E I)メッセージを提案する。実施例は、例えば、前述のフレームワークを使用することができる3次元(3D)モニタ上のマルチビュー・ビデオ・ストリームを簡単かつ便利に表示することを意図している。この概念は、高位構文を使用して前述の情報をシグナリングする他のビデオ符号化の標準及び勧告に拡張することが可能である。

30

【0033】

更に、一実施例では、マルチビュー・ビデオ符号化器及び/復号化器に送出される前にビューをどのようにして配置するかというシグナリング方法を提案している。効果的には、上記実施例は、マルチビュー符号化の単純化された実現形態につながり得、符号化効率に便益をもたらし得る。特定のビューを集約し、疑似ビュー又はスーパーフラグを形成することが可能であり、タイリングされたスーパービューは、例えば、マルチビュー・ビデオ符号化の現在のM P E G - 4 A V C 標準ベースの実現形態に従って、共通のマルチビュー・ビデオ符号化器及び/又は復号化器によって通常ビューとして扱われる。疑似ビューの手法の使用をシグナリングするためにマルチビュー・ビデオ符号化のシーケンス・パラメータ・セット(S P S)において、新たなフラグが、提案されている。実施例は、前述のフレームワークを使用することができる3Dモニタ上のマルチビュー・ビデオ・ストリームを簡単かつ便利に表示することを意図している。

40

【0034】

単一ビューのビデオ符号化/復号化の標準/勧告を使用した符号化/復号化

国際標準化機構/国際電気標準会議(I S O / I E C) 動画像専門家グループ-4(M P E G - 4) パート10 高度ビデオ符号化(A V C) 標準/国際電気通信連合(I T U - T) H . 2 6 4 勧告(以下、「M P E G - 4 A V C 標準」に基づいたマルチビデオ符号

50

化 (M V C) の現在の実現形態では、参照ソフトウェアは、単一の符号化器で各ビューを符号化し、ビュー間の参照を考慮に入れることにより、マルチビュー予測を達成する。各ビューは、別個のビットストリームとして符号化器により、その元の分解能で符号化され、その後、全てのビットストリームを合成して単一のビットストリームを形成する。上記単一のビットストリームはその後、復号化される。各ビューは、別個の Y U V 復号化出力を生成する。

【 0 0 3 5 】

マルチビュー予測の別の手法は、実施可能なダウンサンプリング動作による、より大きなフレーム又はスーパー・フレーム上の (同時にサンプリングされた) 各ビューからのピクチャをタイリングすることを伴う。図 1 に移れば、単一のフレーム上にタイリングされた 4 つのビューの例の全体を参照符号 1 0 0 で示す。図 2 に移れば、単一のフレーム上で反転させ、タイリングされた 4 つのビューの例の全体を参照符号 2 0 0 で示す。よって、4 つのビューが存在している場合、各ビューからのピクチャは、タイルのようなスーパー・フレームに構成される。これは、高分解能の単一の非符号化入力系列をもたらす。前述の信号は次いで、 I S O / I E C M P E G - 2 標準や M P E G - A V C 標準などの既存のビデオ符号化標準を使用して符号化することが可能である。

10

【 0 0 3 6 】

マルチビュー予測の更に別の手法は、新たな標準を使用して、別々のビューを別個に符号化し、復号化後、プレイヤーによって要求されるようにビューをタイリングすることを伴うに過ぎない。

20

【 0 0 3 7 】

多くのディスプレイ製造業者は、単一のフレーム上に別々のビューを配置又はタイリングし、それらのそれぞれの位置からのビューを抽出し、レンダリングする前述のフレームワークを使用する。前述の場合、ビットストリームが前述の特性を有するかを判定する標準的なやり方は存在しない。よって、システムが、大きなフレームにおいて別々のビューのピクチャをタイリングする方法を使用する場合、別々のビューを抽出する方法は、固有のものである。

【 0 0 3 8 】

図 3 を参照すれば、 M P E G - 4 A V C 標準によってビデオ符号化を行うことができるビデオ符号化器は全体を参照符号 3 0 0 で示す。

30

【 0 0 3 9 】

ビデオ符号化器 3 0 0 は、合成器 3 8 5 の非反転入力と信号通信する出力を有するフレーム配列バッファ 3 1 0 を含む。合成器 3 8 5 の出力は変換器及び量子化器 3 2 5 の第 1 の入力と信号通信で接続される。変換器及び量子化器 3 2 5 の出力は、エントロピ符号化器 3 4 5 の第 1 の入力、並びに、逆変換器及び逆量子化器 3 5 0 の第 1 の入力と信号通信で接続される。エントロピ符号化器 3 4 5 の出力は合成器 3 9 0 の第 1 の非反転入力と信号通信で接続される。合成器 3 9 0 の出力は、出力バッファ 3 3 5 の第 1 の入力と信号通信で接続される。

【 0 0 4 0 】

符号化器制御器 3 0 5 の第 1 の出力は、フレーム配列バッファ 3 1 0 の第 2 の入力と、逆変換器及び逆量子化器 3 5 0 の第 2 の入力と、ピクチャ・タイプ決定モジュール 3 1 5 の入力と、マクロブロックタイプ (M B タイプ) 決定モジュール 3 2 0 の入力と、イントラ予測モジュール 3 6 0 の第 2 の入力と、デブロッキング・フィルタ 3 6 5 の第 2 の入力と、動き補償器 3 7 0 の第 1 の入力と、動き補償器 3 7 5 の第 1 の入力と、参照ピクチャ・バッファ 3 8 0 の第 2 の入力と信号通信で接続される。

40

【 0 0 4 1 】

符号化器制御器 3 0 5 の第 2 の出力は、付加拡張情報 (S E I) 挿入器 3 3 0 の第 1 の入力、変換器及び量子化器 3 2 5 の第 2 の入力、エントロピ符号化器 3 4 5 の第 2 の入力、出力バッファ 3 3 5 の第 2 の入力、並びに、シーケンス・パラメータ・セット (S P S) 及びピクチャ・パラメータ・セット (P P S) 挿入器 3 4 0 の入力と信号通信で接続さ

50

れる。

【 0 0 4 2 】

ピクチャ・タイプ決定モジュール 3 1 5 の第 1 の出力は、フレーム配列バッファ 3 1 0 の第 3 の入力と信号通信で接続される。ピクチャ・タイプの決定モジュール 3 1 5 の第 2 の出力は、マクロブロックタイプの決定モジュール 3 2 0 の第 2 の入力と信号通信で接続される。

【 0 0 4 3 】

シーケンス・パラメータ・セット (S P S) 及びピクチャ・パラメータ・セット (P P S) 挿入器 3 4 0 の出力は、合成器 3 9 0 の第 3 の非反転入力と信号通信で接続される。 S E I 挿入器 3 3 0 の出力は、合成器 3 9 0 の第 2 の非反転入力と信号通信で接続される。

10

【 0 0 4 4 】

逆量子化器及び逆変換器 3 5 0 の出力は合成器 3 1 9 の第 1 の非反転入力と信号通信で接続される。合成器 3 1 9 の出力は、イントラ予測モジュール 3 6 0 の第 1 の入力及びデブロッキング・フィルタ 3 6 5 の第 1 の入力と信号通信で接続される。デブロッキング・フィルタ 3 6 5 の出力は参照ピクチャ・バッファ 3 8 0 の第 1 の入力と信号通信で接続される。参照ピクチャ・バッファ 3 8 0 の出力は、動き推定器 3 7 5 の第 2 の入力及び動き補償器 3 7 0 の第 1 の入力と信号通信で接続される。動き推定器 3 7 5 の第 1 の出力は動き補償器 3 7 0 の第 2 の入力と信号通信で接続される。動き推定器 3 7 5 の第 2 の出力はエントロピ符号化器 3 4 5 の第 3 の入力と信号通信で接続される。

20

【 0 0 4 5 】

動き補償器 3 7 0 の出力はスイッチ 3 9 7 の第 1 の入力と信号通信で接続される。イントラ予測モジュール 3 6 0 の出力はスイッチ 3 9 7 の第 2 の入力と信号通信で接続される。マクロブロックタイプの決定モジュール 3 2 0 の出力は、スイッチ 3 9 7 に制御入力を供給するために、スイッチ 3 9 7 の第 3 の入力と信号通信で接続される。スイッチ 3 9 7 の第 3 の入力は、(制御入力、すなわち、第 3 の入力と比較して) スwitch の「データ」入力を動き補償器 3 7 0 又はイントラ予測モジュール 3 6 0 によって供給されるか否かを判定する。スイッチ 3 9 7 の出力は、合成器 3 1 9 の第 2 の非反転入力、及び合成器 3 8 5 の反転入力と信号通信で接続される。

【 0 0 4 6 】

フレーム配列バッファ 3 1 0 及び符号化器制御器 1 0 5 の入力は、入力ピクチャ 3 0 1 を受け取るために、符号化器 3 0 0 の入力として利用可能である。更に、付加拡張情報 (S E I) 挿入器 3 3 0 の入力は、メタデータを受け取るために、符号化器 3 0 0 の入力として利用可能である。出力バッファ 3 3 5 の出力は、ビットストリームを出力するために、符号化器 3 0 0 の出力として利用可能である。

30

【 0 0 4 7 】

図 4 に移れば、 M P E G - 4 A V C 標準によってビデオ復号化を行うことができるビデオ復号化器は全体を参照符号 4 0 0 で示す。

【 0 0 4 8 】

ビデオ復号化器 4 0 0 は、エントロピ復号化器 4 4 5 の第 1 の入力と信号通信で接続された出力を有する入力バッファ 4 1 0 を含む。エントロピ復号化器 4 4 5 の第 1 の出力は、逆変換器及び逆量子化器 4 5 0 の第 1 の入力と信号通信で接続される。逆変換器及び逆量子化器 4 5 0 の出力は合成器 4 2 5 の第 2 の非反転入力と信号通信で接続される。合成器 4 2 5 の出力はデブロッキング・フィルタ 4 6 5 の第 2 の入力及びイントラ予測モジュール 2 5 0 の第 1 の入力と信号通信で接続される。デブロッキング・フィルタ 4 6 5 の第 2 の出力は、参照ピクチャ・バッファ 4 8 0 の第 1 の入力と信号通信で接続される。

40

【 0 0 4 9 】

エントロピ復号化器 4 4 5 の第 2 の出力は、動き補償器 4 7 0 の第 3 の入力及びデブロッキング・フィルタ 4 6 5 の第 1 の入力と信号通信で接続される。エントロピ復号化器 4 4 5 の第 3 の出力は、復号化器制御器 4 0 5 の入力と信号通信で接続される。復号化器制

50

御器 405 の第 1 の出力はエントロピ復号化器 445 の第 2 の入力と信号通信で接続される。復号化器制御器 405 の第 2 の出力は、逆変換器及び逆量子化器 450 の第 2 の入力と信号通信で接続される。復号化器制御器 405 の第 3 の出力は、デブロッキング・フィルタ 465 の第 3 の入力と信号通信で接続される。復号化器制御器 405 の第 4 の出力は、イントラ予測モジュール 460 の第 2 の入力、動き補償器 470 の第 1 の入力、及び参照ピクチャ・バッファ 480 の第 2 の入力と信号通信で接続される。

【0050】

動き補償器 470 の出力は、スイッチ 497 の第 1 の入力と信号通信で接続される。イントラ予測モジュール 460 の出力はスイッチ 497 の第 2 の入力と信号通信で接続される。スイッチ 497 の出力は合成器 425 の第 1 の非反転入力と信号通信で接続される。

10

【0051】

入力バッファ 410 の入力は、入力ビットストリームを受け取るために、復号化器 400 の入力として利用可能である。デブロッキング・フィルタ 465 の第 1 の出力は、出力ピクチャを出力するために、復号化器 400 の出力として利用可能である。

【0052】

図 5 に移れば、MPEG-4 AVC 標準を使用して複数のビューのピクチャを符号化する例示的な方法の全体を参照符号 500 で示す。

【0053】

方法 500 は、開始ブロック 502 を含む。開始ブロック 502 は制御を機能ブロック 504 に渡す。機能ブロック 504 は、特定の時点における各ビューを、タイル形式におけるサブピクチャとして配置し、制御を機能ブロック 506 に渡す。機能ブロック 506 は構文要素 `num_coded_views_minus_1` をセットし、制御を機能ブロック 508 に渡す。機能ブロック 508 は構文要素 `org_pic_width_in_mbs_minus_1` 及び `org_pic_height_in_mbs_minus_1` をセットし、制御を機能ブロック 510 に渡す。機能ブロック 510 は変数 `i` をゼロに等しくセットし、制御を決定ブロック 512 に渡す。決定ブロック 512 は、変数 `i` がビュー数未満であるか否かを判定する。肯定の場合、制御を機能ブロック 514 に渡す。さもなければ、制御を機能ブロック 524 に渡す。

20

【0054】

機能ブロック 514 は、構文要素 `view_id[i]` をセットし、制御を機能ブロック 516 に渡す。機能ブロック 516 は構文要素 `num_parts(view_id[i])` をセットし、制御を機能ブロック 518 に渡す。機能ブロック 518 は変数 `j` をゼロにセットし、制御を決定ブロック 520 に渡す。決定ブロック 520 は、変数 `j` の現在値が、構文要素 `num_parts(view_id[i])` の現在値未満か否かを判定する。肯定の場合、制御を機能ブロック 522 に渡す。さもなければ、制御を機能ブロック 528 に渡す。

30

【0055】

機能ブロック 522 は、

`depth_flag[view_id[i]][j];`

`flip_dir[view_id[i]][j];`

`loc_left_offset[view_id[i]][j];`

`loc_top_offset[view_id[i]][j];`

`frame_crop_left_offset[view_id[i]][j];`

`frame_crop_right_offset[view_id[i]][j];`

`frame_crop_top_offset[view_id[i]][j];` 及び

40

`frame_crop_bottom_offset[view_id[i]][j]`

の構文要素をセットし、変数 `j` を増やし、次いで制御を決定ブロック 520 に戻す。

【0056】

機能ブロック 528 は構文要素 `upsample_view_flag[view_i`

50

`d[i]` をセットし、制御を決定ブロック 530 に渡す。決定ブロック 530 は、構文要素 `upsample_view_flag[view_id[i]]` の現在値が 1 に等しいか否かを判定する。肯定の場合、制御を機能ブロック 532 に渡す。さもなければ、制御を決定ブロック 534 に渡す。

【0057】

機能ブロック 532 は構文要素 `upsample_filter[view_id[i]]` をセットし、制御を決定ブロック 534 に渡す。

【0058】

決定ブロック 534 は、構文要素 `upsample_filter[view_id[i]]` の現在値が 3 に等しいか否かを判定する。肯定の場合、制御を機能ブロック 536 に渡す。さもなければ、制御を機能ブロック 540 に渡す。

10

【0059】

機能ブロック 536 は、

```
vert_dim[view_id[i]];
hor_dim[view_id[i]]; 及び
quantizer[view_id[i]]
```

の構文要素をセットし、制御を機能ブロック 538 に渡す。機能ブロック 538 は、YUV 成分毎にフィルタ係数をセットし、制御を機能ブロック 540 に渡す。

【0060】

機能ブロック 540 は変数 `i` を増やし、制御を決定ブロック 512 に戻す。

20

【0061】

機能ブロック 524 は、前述の構文要素を、シーケンス・パラメータ・セット (SPS)、ピクチャ・パラメータ・セット (PPS)、付加拡張情報 (SEI) メッセージ、ネットワーク抽象化層 (NAL) ユニット・ヘッダ、及びスライス・ヘッダのうち少なくとも 1 つに書き込み、制御を機能ブロック 526 に渡す。機能ブロック 526 は、各ピクチャを MPEG-4 AVC 標準又は他の単一ビュー・コーデックを使用して符号化し、制御を終了ブロック 599 に渡す。

【0062】

図 6 に移れば、MPEG-4 AVC 標準を使用して複数のビューのピクチャを復号化する例示的な方法の全体を参照符号 600 で示す。

30

【0063】

方法 600 は、開始ブロック 602 を含む。開始ブロック 602 は制御を機能ブロック 604 に渡す。機能ブロック 604 は、シーケンス・パラメータ・セット (SPS)、ピクチャ・パラメータ・セット (PPS)、付加拡張情報 (SEI) メッセージ、ネットワーク抽象化 (NAL) ユニット・ヘッダ、及びスライス・ヘッダのうち少なくとも 1 つからの以下の構文要素を解析し、制御を機能ブロック 606 に渡す。機能ブロック 606 は、構文要素 `num_coded_views_minus1` を解析し、制御を機能ブロック 608 に渡す。機能ブロック 608 は、構文要素 `org_pic_width_in_mbs_minus1` 及び `org_pic_height_in_mbs_minus1` を解析し、制御を機能ブロック 610 に渡す。機能ブロック 610 は変数 `i` をゼロに等しくセットし、制御を決定ブロック 612 に渡す。決定ブロック 612 は、変数 `i` がビュー数未満であるか否かを判定する。肯定の場合、制御を機能ブロック 614 に渡す。さもなければ、制御を機能ブロック 624 に渡す。

40

【0064】

機能ブロック 614 は、構文要素 `view_id[i]` を解析し、制御を機能ブロック 616 に渡す。機能ブロック 616 は、構文要素 `num_parts_minus1[view_id[i]]` を解析し、制御を機能ブロック 618 に渡す。機能ブロック 618 は変数 `j` をゼロに等しくセットし、制御を決定ブロック 620 に渡す。決定ブロック 620 は、変数 `j` の現在値が構文要素 `num_parts[view_id[i]]` の現在値未満であるか否かを判定する。肯定の場合、制御を機能ブロック 622 に渡す。さもな

50

れば、制御を機能ブロック628に渡す。

【0065】

機能ブロック622は、

```
depth_flag[view_id[i]][j];
flip_dir[view_id[i]][j];
loc_left_offset[view_id[i]][j];
loc_top_offset[view_id[i]][j];
frame_crop_left_offset[view_id[i]][j];
frame_crop_right_offset[view_id[i]][j];
frame_crop_top_offset[view_id[i]][j]; 及び
```

び

```
frame_crop_bottom_offset[view_id[i]][j]
```

の構文要素を解析し、変数jを増やし、次いで、制御を決定ブロック620に戻す。

【0066】

機能ブロック628は構文要素upsample_view_flag[view_id[i]]を解析し、制御を決定ブロック630に渡す。決定ブロック630は、構文要素upsample_view_flag[view_id[i]]の現在値が1に等しいか否かを判定する。肯定の場合、制御を機能ブロック632に渡す。さもなければ、制御を決定ブロック634に渡す。

【0067】

機能ブロック632は構文要素upsample_filter[view_id[i]]を解析し、制御を決定ブロック634に渡す。

【0068】

決定ブロック634は、構文要素upsample_filter[view_id[i]]の現在値が3に等しいか否かを判定する。肯定の場合、制御を機能ブロック636に渡す。さもなければ、制御を機能ブロック640に渡す。

【0069】

機能ブロック636は、

```
vert_dim[view_id[i]];
hor_dim[view_id[i]]; 及び quantizer[view_ 30
id[i]]
```

の構文要素を解析し、制御を機能ブロック638に渡す。

【0070】

機能ブロック638は、YUV成分毎にフィルタ係数を解析し、制御を機能ブロック640に渡す。

【0071】

機能ブロック640は、変数iを増やし、制御を決定ブロック612に戻す。

【0072】

機能ブロック624は、MPEG-4 AVC標準又は他の単一ビュー・コーデックを使用して各ピクチャを復号化し、制御を機能ブロック626に渡す。機能ブロック626は各ビューをピクチャから、高位構文を使用して分離し、制御を終了ブロック699に渡す。

【0073】

図7に移れば、MPEG-4 AVC標準を使用して複数のビュー及び深度のピクチャを符号化する例示的な方法の全体を参照符号700で示す。

【0074】

方法700は、開始ブロック702を含む。開始ブロック702は制御を機能ブロック704に渡す。機能ブロック704は、特定の時点における各ビュー及び対応する深度を、タイル形式におけるサブピクチャとして配置し、制御を機能ブロック706に渡す。機能ブロック706は構文要素num_coded_views_minus1をセットし、制御を機能ブロック708に渡す。機能ブロック708は構文要素org_pic_w 50

`id th__in__mbs__minus 1`及び`org__pic__height__in__mbs__minus 1`をセットし、制御を機能ブロック710に渡す。機能ブロック710は変数*i*をゼロに等しくセットし、制御を決定ブロック712に渡す。決定ブロック712は、変数*i*がビュー数未満であるか否かを判定する。肯定の場合、制御を機能ブロック714に渡す。さもなければ、制御を機能ブロック724に渡す。

【0075】

機能ブロック714は、構文要素`view__id[i]`をセットし、制御を機能ブロック716に渡す。機能ブロック716は、構文要素`num__parts[view__id[i]]`をセットし、制御を機能ブロック718に渡す。機能ブロック718は、変数*j*をゼロに等しくセットし、制御を決定ブロック720に渡す。決定ブロック720は、変数*j*の現在値が構文要素`num__parts[view__id[i]]`の現在値未満であるか否かを判定する。肯定の場合、制御を機能ブロック722に渡す。さもなければ、制御を機能ブロック728に渡す。

10

【0076】

機能ブロック722は、`depth__flag[view__id[i]][j]`;
`flip__dir[view__id[i]][j]`;
`loc__left__offset[view__id[i]][j]`;
`loc__top__offset[view__id[i]][j]`;
`frame__crop__left__offset[view__id[i]][j]`;
`frame__crop__right__offset[view__id[i]][j]`;
`frame__crop__top__offset[view__id[i]][j]`; 及

20

び

`frame__crop__bottom__offset[view__id[i]][j]`の構文要素をセットし、変数*j*を増やし、次いで制御を決定ブロック720に戻す。

【0077】

機能ブロック728は構文要素`upsample__view__flag[view__id[i]]`をセットし、制御を決定ブロック730に渡す。決定ブロック730は、構文要素`upsample__view__flag[view__id[i]]`の現在値が1に等しいか否かを判定する。肯定の場合、制御を機能ブロック732に渡す。さもなければ、制御を決定ブロック734に渡す。

30

【0078】

機能ブロック732は構文要素`upsample__filter[view__id[i]]`をセットし、制御を決定ブロック734に渡す。

【0079】

決定ブロック734は、構文要素`upsample__filter[view__id[i]]`の現在値が3に等しいか否かを判定する。肯定の場合、制御を機能ブロック736に渡す。さもなければ、制御を機能ブロック740に渡す。

【0080】

機能ブロック736は、
`vert__dim[view__id[i]]`;
`hor__dim[view__id[i]]`; 及び
`quantizer[view__id[i]]`

40

の構文要素をセットし、制御を機能ブロック738に渡す。

【0081】

機能ブロック738は、YUV成分毎にフィルタ係数をセットし、制御を機能ブロック740に渡す。

【0082】

機能ブロック740は、変数*i*を増やし、制御を決定ブロック712に戻す。

【0083】

機能ブロック724は前述の構文要素を、シーケンス・パラメータ・セット(SPS)

50

、ピクチャ・パラメータ・セット (P P S)、付加拡張情報 (S E I) メッセージ、ネットワーク抽象化層 (N A L) ユニット・ヘッダ、及びスライス・ヘッダのうちの少なくとも一つに書き込み、制御を機能ブロック 7 2 6 に渡す。機能ブロック 7 2 6 は、M P E G - 4 A V C 標準や他の単一ビュー・コーデックを使用して各ピクチャを符号化し、制御を終了ブロック 7 9 9 に渡す。

【 0 0 8 4 】

図 8 に移れば、M P E G - 4 A V C 標準を使用して複数のビュー及び深度のピクチャを復号化する例示的な方法の全体を参照符号 8 0 0 で示す。

【 0 0 8 5 】

方法 8 0 0 は、開始ブロック 8 0 2 を含む。開始ブロック 8 0 2 は制御を機能ブロック 8 0 4 に渡す。機能ブロック 8 0 4 は、シーケンス・パラメータ・セット (S P S)、ピクチャ・パラメータ・セット (P P S)、付加拡張情報 (S E I) メッセージ、ネットワーク抽象化 (N A L) ユニット・ヘッダ、及びスライス・ヘッダのうちの少なくとも一つからの以下の構文要素を解析し、制御を機能ブロック 8 0 6 に渡す。機能ブロック 8 0 6 は、構文要素 `num__coded__views__minus 1` を解析し、制御を機能ブロック 8 0 8 に渡す。機能ブロック 8 0 8 は、構文要素 `org__pic__width__in__mbs__minus 1` 及び `org__pic__height__in__mbs__minus 1` を解析し、制御を機能ブロック 8 1 0 に渡す。機能ブロック 8 1 0 は変数 `i` をゼロに等しくセットし、制御を決定ブロック 8 1 2 に渡す。決定ブロック 8 1 2 は、変数 `i` がビュー数未満であるか否かを判定する。肯定の場合、制御を機能ブロック 8 1 4 に渡す。さもなければ、制御を機能ブロック 8 2 4 に渡す。

10

20

【 0 0 8 6 】

機能ブロック 8 1 4 は、構文要素 `view__id [i]` を解析し、制御を機能ブロック 8 1 6 に渡す。機能ブロック 8 1 6 は、構文要素 `num__parts__minus 1 [view__id [i]]` を解析し、制御を機能ブロック 8 1 8 に渡す。機能ブロック 8 1 8 は変数 `j` をゼロに等しくセットし、制御を決定ブロック 8 2 0 に渡す。決定ブロック 8 2 0 は、変数 `j` の現在値が構文要素 `num__parts [view__id [i]]` の現在値未満であるか否かを判定する。肯定の場合、制御を機能ブロック 8 2 2 に渡す。さもなければ、制御を機能ブロック 8 2 8 に渡す。

30

【 0 0 8 7 】

機能ブロック 8 2 2 は、

```
depth__flag [ view__id [ i ] ] [ j ] ;
flip__dir [ view__id [ i ] ] [ j ] ;
loc__left__offset [ view__id [ i ] ] [ j ] ;
loc__top__offset [ view__id [ i ] ] [ j ] ;
frame__crop__left__offset [ view__id [ i ] ] [ j ] ;
frame__crop__right__offset [ view__id [ i ] ] [ j ] ;
frame__crop__top__offset [ view__id [ i ] ] [ j ] ; 及
```

び

```
frame__crop__bottom__offset [ view__id [ i ] ] [ j ]
```

40

の構文要素をセットし、変数 `j` を増やし、次いで制御を決定ブロック 8 2 0 に戻す。

【 0 0 8 8 】

機能ブロック 8 2 8 は構文要素 `upsample__view__flag [view__id [i]]` をセットし、制御を決定ブロック 8 3 0 に渡す。決定ブロック 8 3 0 は、構文要素 `upsample__view__flag [view__id [i]]` の現在値が 1 に等しいか否かを判定する。肯定の場合、制御を機能ブロック 8 3 2 に渡す。さもなければ、制御を決定ブロック 8 3 4 に渡す。

【 0 0 8 9 】

機能ブロック 8 3 2 は構文要素 `upsample__filter [view__id [i]]` を解析し、制御を決定ブロック 8 3 4 に渡す。

50

【0090】

決定ブロック834は、構文要素 `upsample_filter[view_id[i]]` の現在値が3に等しいか否かを判定する。肯定の場合、制御を機能ブロック836に渡す。さもなければ、制御を機能ブロック840に渡す。

【0091】

機能ブロック836は、

```
vert_dim[view_id[i]];
hor_dim[view_id[i]]; 及び
quantizer[view_id[i]]
```

の構文要素を解析し、制御を機能ブロック838に渡す。

10

【0092】

機能ブロック838は、YUV成分毎にフィルタ係数を解析し、制御を機能ブロック840に渡す。

【0093】

機能ブロック840は変数 `i` を増やし、制御を決定ブロック812に戻す。

【0094】

機能ブロック824は、MPEG-4 AVC標準又は他の単一ビュー・コーデックを使用して各ピクチャを復号化し、制御を機能ブロック826に渡す。機能ブロック826は、各ビュー及び対応する深度をピクチャから高位構文を使用して分離し、制御を機能ブロック827に渡す。機能ブロック827は潜在的には、抽出されたビュー信号及び深度信号を使用してビュー合成を行い、制御を終了ブロック899に渡す。

20

【0095】

図7及び図8において使用した深度に関して、図9は、深度信号900の例を示す(深度は、画像(図示せず)の対応する場所毎の画素値として備えられる)。更に、図10は、タイル1000に含まれた2つの深度信号の例を示す。タイル1000の右上部分は、タイル1000の左上の画像に対応する深度値を有する深度信号である。タイル1000の右下部分は、タイル1000の左下の画像に対応する深度値を有する深度信号である。

【0096】

図11に移れば、単一のフレーム上にタイリングされた5つのビューの例の全体を参照符号1100で示す。上の4つのビューは全て、通常の向きにある。5番目のビューも通常の向きにあるが、タイル1100の底部に沿って2つの部分に分割されている。5番目のビューの左部分は5番目のビューの「最上部」を示し、5番目のビューの右部分は5番目のビューの「底部」を示す。

30

【0097】

マルチビュー・ビデオ符号化/復号化の標準/勧告を使用した符号化/復号化

図12に移れば、例示的なマルチビュー・ビデオ符号化(MVC)符号化器の全体を参照符号1200で示す。符号化器1200は、変換器1210の入力と信号通信で接続された出力を有する合成器1205を含む。変換器1210の出力は、量子化器1215の入力と信号通信で接続される。量子化器1215の出力はエントロピ符号化器1220の入力及び逆量子化器1225の入力と信号通信で接続される。逆量子化器1225の出力は逆変換器1230の入力と信号通信と接続される。逆変換器1230の出力は合成器1235の第1の非反転入力と信号通信で接続される。合成器1235の出力はイントラ予測器1245の入力及びデブロッキング・フィルタ1250の入力と信号通信で接続される。デブロッキング・フィルタ1250の出力は、(ビュー `i` の)参照ピクチャ記憶装置1255の入力と信号通信で接続される。参照ピクチャ記憶装置1255の出力は、動き補償器1275の第1の入力及び動き推定器1280の第1の入力と信号通信で接続される。動き推定器1280の出力は、動き補償器1275の第2の入力と信号通信で接続される。(他のビューの)参照ピクチャ記憶装置1260の出力は、視差推定器1270の第1の入力及び視差補償器1265の第1の入力と信号通信で接続される。視差推定器1270の出力は、視差補償器1265の第2の入力と信号通信で接続される。

40

50

【 0 0 9 8 】

エンプロピ復号化器 1 2 2 0 の出力は符号化器 1 2 0 0 の出力として利用可能である。合成器 1 2 0 5 の非反転入力、符号化器 1 2 0 0 の入力として利用可能であり、視差推定器 1 2 7 0 の第 2 の入力及び動き推定器 1 2 8 0 の第 2 の入力と信号通信で接続される。スイッチ 1 2 8 5 の出力は、合成器 1 2 3 5 の第 2 の非反転入力及び合成器 1 2 0 5 の反転入力と信号通信で接続される。スイッチ 1 2 8 5 は、動き補償器 1 2 7 5 の出力と信号通信で接続された第 1 の入力と、視差補償器 1 2 6 5 の出力と信号通信で接続された第 2 の入力と、イントラ予測器 1 2 4 5 の出力と信号通信で接続された第 3 の入力を含む。

【 0 0 9 9 】

モード決定モジュール 1 2 4 0 は、スイッチ 1 2 8 5 によって選択される入力がどれであるかを制御するために、スイッチ 1 2 8 5 に接続された出力を有する。

【 0 1 0 0 】

図 1 3 に移れば、例示的なマルチビュー・ビデオ符号化 (M V C) 復号化器の全体を参照符号 1 3 0 0 で示す。復号化器 1 3 0 0 は、逆量子化器 1 3 1 0 の入力と信号通信で接続された出力を有するエンプロピ復号化器 1 3 0 5 を含む。逆量子化器の出力は、逆変換器 1 3 1 5 の入力と信号通信で接続される。逆変換器 1 3 1 5 の出力は合成器 1 3 2 0 の第 1 の非反転入力と信号通信で接続される。合成器 1 3 2 0 の出力は、デブロッキング・フィルタ 1 3 2 5 の入力及びイントラ予測器 1 3 3 0 の入力と信号通信で接続される。デブロッキング・フィルタ 1 3 2 5 の出力は、(ビュー *i* の) 参照ピクチャ記憶装置 1 3 4 0 の入力と信号通信で接続される。参照ピクチャ記憶装置 1 3 4 0 の出力は、動き補償器 1 3 3 5 の第 1 の入力と信号通信で接続される。

【 0 1 0 1 】

(他のビューの) 参照ピクチャ記憶装置 1 3 4 5 の出力は、視差補償器 1 3 5 0 の第 1 の入力と信号通信で接続される。

【 0 1 0 2 】

エンプロピ符号化器 1 3 0 5 の入力、残差ビットストリームを受け取るために、復号化器 1 3 0 0 の入力として利用可能である。更に、エンプロピ符号化器 1 3 6 0 の入力、スイッチ 1 3 5 5 によって選択される入力がどれであるかを制御するために制御構文を受け取るために、復号化器 1 3 0 0 への入力としても利用可能である。更に、動き補償器 1 3 5 5 の第 2 の入力、動きベクトルを受け取るために、復号化器 1 3 0 0 の入力として利用可能である。更に、視差補償器 1 3 5 0 の第 2 の入力、視差ベクトルを受け取るために、復号化器 1 3 0 0 への入力として利用可能である。

【 0 1 0 3 】

スイッチ 1 3 5 5 の出力は合成器 1 3 2 0 の第 2 の非反転入力と信号通信で接続される。スイッチ 1 3 5 5 の第 1 の入力は視差補償器 1 3 5 0 の出力と信号通信で接続される。スイッチ 1 3 5 5 の第 2 の入力は、動き補償器 1 3 3 5 の出力と信号通信で接続される。スイッチ 1 3 5 5 の第 3 の入力は、イントラ予測器 1 3 3 0 の出力と信号通信で接続される。モード・モジュール 1 3 6 0 の出力は、スイッチ 1 3 5 5 によって選択される入力がどれであるかを制御するために、スイッチ 1 3 5 5 と信号通信で接続される。デブロッキング・フィルタ 1 3 2 5 の出力は、復号化器 1 3 0 0 の出力として利用可能である。

【 0 1 0 4 】

図 1 4 に移れば、MPEG-4 AVC 標準のマルチビュー・ビデオ符号化 (M V C) の拡張を使用したピクチャの符号化に備えて複数のビューのピクチャを処理する例示的な方法の全体を参照符号 1 4 0 0 で示す。

【 0 1 0 5 】

方法 1 4 0 0 は、開始ブロック 1 4 0 5 を含む。開始ブロック 1 4 0 5 は制御を機能ブロック 1 4 1 0 に渡す。機能ブロック 1 4 1 0 は、特定の時点で、合計 M 個のビューのうち N 個のビュー毎に、タイル形式におけるスーパー・ピクチャとして配置し、制御を機能ブロック 1 4 1 5 に渡す。機能ブロック 1 4 1 5 は、構文要素 `num_coded_v`

10

20

30

40

50

`views_minus_1`をセットし、制御を機能ブロック1420に渡す。機能ブロック1420は、 $(\text{num_coded_views_minus_1} + 1)$ 個のビュー全てについて構文要素`view_id[i]`をセットし、制御をブロック1425に渡す。機能ブロック1425はアンカ・ピクチャのビュー間参照依存性情報をセットし、制御を機能ブロック1430に渡す。機能ブロック1430は、非アンカ・ピクチャのビュー間参照依存性情報をセットし、制御を機能ブロック1435に渡す。機能ブロック1435は、構文要素`pseudo_view_present_flag`をセットし、制御を決定ブロック1440に渡す。決定ブロック1440は、構文要素`pseudo_view_present_flag`の現在値が`true`に等しいか否かを判定する。肯定の場合、制御を機能ブロック1445に渡す。さもなければ、制御を終了ブロック1499に渡す。

10

【0106】

機能ブロック1445は、

```
tiling_mode;
```

```
org_pic_width_in_mbs_minus_1; 及び
```

```
org_pic_height_in_mbs_minus_1
```

の構文要素をセットし、制御を機能ブロック1450に渡す。機能ブロック1450は、符号化ビュー毎に構文要素`pseudo_view_info(view_id)`を呼び出し、制御を終了ブロック1499に渡す。

20

【0107】

図15に移れば、MPEG-4 AVC標準のマルチビュー・ビデオ符号化(MVC)の拡張を使用して複数のビューのピクチャを符号化する例示的な方法の全体を参照符号1500で示す。

【0108】

方法1500は開始ブロック1502を含む。開始ブロック1502は、入力パラメータ`pseudo_view_id`を有し、制御を機能ブロック1504に渡す。機能ブロック1504は、構文要素`num_sub_views_minus_1`をセットし、制御を機能ブロック1506に渡す。機能ブロック1506は、変数`i`をゼロに等しくセットし、制御を決定ブロック1508に渡す。決定ブロック1508は、変数`i`が`sub_views`の数未満であるか否かを判定する。肯定の場合、制御を機能ブロック1510に渡す。さもなければ、制御を機能ブロック1520に渡す。

30

【0109】

機能ブロック1510は、構文要素`sub_view_id[i]`をセットし、制御を機能ブロック1512に渡す。機能ブロック1512は、構文要素`num_parts_minus_1[sub_view_id[i]]`をセットし、制御を機能ブロック1514に渡す。機能ブロック1514は変数`j`をゼロに等しくセットし、制御を決定ブロック1516に渡す。決定ブロック1516は、変数`j`が構文要素`num_parts[sub_view_id[i]]`未満であるか否かを判定する。肯定の場合、制御を機能ブロック1518に渡す。さもなければ、制御を決定ブロック1522に渡す。

40

【0110】

機能ブロック1518は、

```
loc_left_offset[sub_view_id[i]][j];
```

```
loc_top_offset[sub_view_id[i]][j];
```

```
frame_crop_left_offset[sub_view_id[i]][j];
```

```
frame_crop_right_offset[sub_view_id[i]][j];
```

```
frame_crop_top_offset[sub_view_id[i]][j];
```

及び

50

```
frame_crop_bottom_offset[sub_view_id[i][j]].
```

の構文要素をセットし、変数 j を増やし、制御を決定ブロック 1516 に戻す。

【0111】

機能ブロック 1520 は、マルチビュー・ビデオ符号化 (MVC) を使用して現在のビューの現在のピクチャを符号化し、制御を終了ブロック 1599 に渡す。

【0112】

決定ブロック 1522 は、構文要素 `tiling_mode` がゼロに等しいか否かを判定する。肯定の場合、制御を機能ブロック 1524 に渡す。さもなければ、制御を機能ブロック 1538 に渡す。

【0113】

機能ブロック 1524 は、構文要素 `flip_dir[sub_view_id[i]]` 及び構文要素 `upsample_view_flag[sub_view_id[i]]` をセットし、制御を決定ブロック 1526 に渡す。決定ブロック 1526 は、構文要素 `upsample_view_flag[sub_view_id[i]]` の現在値が 1 に等しいか否かを判定する。肯定の場合、制御を機能ブロック 1528 に渡す。さもなければ、制御を決定ブロック 1530 に渡す。

【0114】

機能ブロック 1528 は構文要素 `upsample_filter[sub_view_id[i]]` をセットし、制御を決定ブロック 1530 に渡す。決定ブロック 1530 は、構文要素 `upsample_filter[sub_view_id[i]]` の値が 3 に等しいか否かを判定する。肯定の場合、制御を機能ブロック 1532 に渡す。さもなければ、制御を機能ブロック 1536 に渡す。

【0115】

機能ブロック 1532 は、

```
vert_dim[sub_view_id[i]];
hor_dim[sub_view_id[i]]; 及び
quantizer[sub_view_id[i]]
```

の構文要素をセットし、制御を機能ブロック 1534 に渡す。機能ブロック 1534 は、YUV 成分毎にフィルタ係数をセットし、制御を機能ブロック 1536 に渡す。

【0116】

機能ブロック 1536 は変数 i を増やし、制御を決定ブロック 1508 に戻す。

【0117】

機能ブロック 1538 は、構文要素 `pixel_dist_x[sub_view_id[i]]` 及び構文要素 `flip_dist_y[sub_view_id[i]]` をセットし、制御を機能ブロック 1540 に渡す。機能ブロック 1540 は変数 j をゼロに等しくセットし、制御を決定ブロック 1542 に渡す。決定ブロック 1542 は、変数 j の現在値が構文要素 `num_parts[sub_view_id[i]]` の現在値未満であるか否かを判定する。肯定の場合、制御を機能ブロック 1544 に渡す。さもなければ、制御を機能ブロック 1536 に渡す。

【0118】

機能ブロック 1544 は構文要素 `num_pixel_tiling_filter_coefs_minus1[sub_view_id[i]]` をセットし、制御を機能ブロック 1546 に渡す。機能ブロック 1546 は、画素タイリング・フィルタ全ての係数をセットし、制御を機能ブロック 1536 に渡す。

【0119】

図 16 に移れば、MPEG-4 AVC 標準のマルチビュー・ビデオ符号化 (MVC) の拡張を使用したピクチャの復号化に備えて複数のビューのピクチャを処理する例示的な方法の全体を参照符号 1600 で示す。

【0120】

10

20

30

40

50

方法1600は、開始ブロック1605を含み、開始ブロック1605は機能ブロック1615に制御を渡す。機能ブロック1615は、構文要素 `num_coded_views_minus_1` を解析し、制御を機能ブロック1620に渡す。機能ブロック1620は、 $(\text{num_coded_views_minus_1} + 1)$ 個のビュー全ての構文要素 `view_id[i]` を解析し、制御を機能ブロック1625に渡す。機能ブロック1625はアンカ・ピクチャのビュー間参照依存性情報を解析し、制御を機能ブロック1630に渡す。機能ブロック1630は非アンカ・ピクチャのビュー間参照依存性情報を解析し、制御を機能ブロック1635に渡す。機能ブロック1635は、構文要素 `pseudo_view_present_flag` を解析し、制御を決定ブロック1640に渡す。決定ブロック1640は、構文要素 `pseudo_view_present_flag` の現在値が `true` に等しいか否かを判定する。肯定の場合、制御を機能ブロック1645に渡す。さもなければ、制御を終了ブロック1699に渡す。

10

【0121】

機能ブロック1645は、
`tiling_mode` ;
`org_pic_width_in_mbs_minus_1` ; 及び
`org_pic_height_in_mbs_minus_1`
 の構文要素を解析し、制御を機能ブロック1650に渡す。機能ブロック1650は、符号化ビュー毎に構文要素 `pseudo_view_info(view_id)` を呼び出し、制御を終了ブロック1699に渡す。

20

【0122】

図17に移れば、MPEG-4 AVC標準のマルチビュー・ビデオ符号化(MVC)の拡張を使用して複数のビューのピクチャを復号化する例示的な方法の全体を参照符号1700で示す。

【0123】

方法1700は開始ブロック1702を含む。開始ブロック1702は、入力パラメータ `pseudo_view_id` から始まり、制御を機能ブロック1704に渡す。機能ブロック1704は、構文要素 `num_sub_views_minus_1` を解析し、制御を機能ブロック1706に渡す。機能ブロック1706は、変数 `i` をゼロに等しくセットし、制御を決定ブロック1708に渡す。決定ブロック1708は、変数 `i` が `sub_views` の数未満であるか否かを判定する。肯定の場合、制御を機能ブロック1710に渡す。さもなければ、制御を機能ブロック1720に渡す。

30

【0124】

機能ブロック1710は、構文要素 `sub_view_id[i]` を解析し、制御を機能ブロック1712に渡す。機能ブロック1712は、構文要素 `num_parts_minus_1[sub_view_id[i]]` を解析し、制御を機能ブロック1714に渡す。機能ブロック1714は変数 `j` をゼロに等しくセットし、制御を決定ブロック1716に渡す。決定ブロック1716は、変数 `j` が構文要素 `num_parts_minus_1[sub_view_id[i]]` 未満であるか否かを判定する。肯定の場合、制御を機能ブロック1718に渡す。さもなければ、制御を決定ブロック1722に渡す。

40

【0125】

機能ブロック1718は、
`loc_left_offset[sub_view_id[i]][j]` ;
`loc_top_offset[sub_view_id[i]][j]` ;
`frame_crop_left_offset[sub_view_id[i]][j]` ;
`frame_crop_right_offset[sub_view_id[i]][j]` ;
`frame_crop_top_offset[sub_view_id[i]][j]` ;

50

];及び

```
frame_crop_bottom_offset[sub_view_id[i]]
][j]
```

の構文要素をセットし、変数 j を増やし、制御を決定ブロック 1716 に戻す。

【0126】

機能ブロック 1720 は、マルチビュー・ビデオ符号化 (MVC) を使用して現在のビューの現在のピクチャを復号化し、制御を機能ブロック 1721 に渡す。機能ブロック 1721 は、各ビューをピクチャから高位構文を使用して分離し、制御を終了ブロック 1799 に渡す。

【0127】

各ビューの復号化ピクチャとの分離は、ビットストリームに示された高位構文を使用して行われる。この高位構文は、ピクチャにおいて存在しているビューの考えられる向き (及び考えられる対応する深度) 並びに厳密な位置を示し得る。

【0128】

決定ブロック 1722 は、構文要素 `tiling_mode` がゼロに等しいか否かを判定する。肯定の場合、制御を機能ブロック 1724 に渡す。さもなければ、制御を機能ブロック 1738 に渡す。

【0129】

機能ブロック 1724 は、構文要素 `flip_dir[sub_view_id[i]]` 及び構文要素 `upsample_view_flag[sub_view_id[i]]` を解析し、制御を決定ブロック 1726 に渡す。決定ブロック 1726 は、構文要素 `upsample_view_flag[sub_view_id[i]]` の現在値が 1 に等しいか否かを判定する。肯定の場合、制御を機能ブロック 1728 に渡す。さもなければ、制御を決定ブロック 1730 に渡す。

【0130】

機能ブロック 1728 は構文要素 `upsample_filter[sub_view_id[i]]` を解析し、制御を決定ブロック 1730 に渡す。決定ブロック 1730 は、構文要素 `upsample_filter[sub_view_id[i]]` の値が 3 に等しいか否かを判定する。肯定の場合、制御を機能ブロック 1732 に渡す。さもなければ、制御を機能ブロック 1736 に渡す。

【0131】

機能ブロック 1732 は、

```
vert_dim[sub_view_id[i]];
hor_dim[sub_view_id[i]];及び
quantizer[sub_view_id[i]]
```

の構文要素を解析し、制御を機能ブロック 1734 に渡す。機能ブロック 1734 は、YUV 成分毎にフィルタ係数を解析し、制御を機能ブロック 1736 に渡す。

【0132】

機能ブロック 1736 は変数 i を増やし、制御を決定ブロック 1708 に戻す。

【0133】

機能ブロック 1738 は、構文要素 `pixel_dist_x[sub_view_id[i]]` 及び構文要素 `flip_dist_y[sub_view_id[i]]` を解析し、制御を機能ブロック 1740 に渡す。機能ブロック 1740 は変数 j をゼロに等しくセットし、制御を決定ブロック 1742 に渡す。決定ブロック 1742 は、変数 j の現在値が構文要素 `num_parts[sub_view_id[i]]` の現在値未満であるか否かを判定する。肯定の場合、制御を機能ブロック 1744 に渡す。さもなければ、制御を機能ブロック 1736 に渡す。

【0134】

機能ブロック 1744 は構文要素 `num_pixel_tiling_filter_coef_minus1[sub_view_id[i]]` を解析し、制御を機能ブ

10

20

30

40

50

ロック1746に渡す。機能ブロック1776は、画素タイリング・フィルタ全ての係数を解析し、制御を機能ブロック1736に渡す。

【0135】

図18に移れば、MPEG-4 AVC標準のマルチビュー・ビデオ符号化(MVC)の拡張を使用したピクチャの符号化に備えて複数のビュー及び深度のピクチャを処理する例示的な方法の全体を参照符号1800で示す。

【0136】

方法1800は開始ブロック1805を含む。開始ブロック1805は制御を機能ブロック1810に渡す。機能ブロック1810は、特定の時点で、合計M個のビュー及び深度マップのうちN個のビュー及び深度マップ毎に、タイル形式におけるスーパー・ピクチャとして配置し、制御を機能ブロック1815に渡す。機能ブロック1815は、構文要素num_coded_views_minus1をセットし、制御を機能ブロック1820に渡す。機能ブロック1820は、view_id[i]に対応する(num_coded_views_minus1 + 1)個の深度全てについて構文要素view_id[i]をセットし、制御を機能ブロック1825に渡す。機能ブロック1825はアンカ深度ピクチャのビュー間参照依存性情報をセットし、制御を機能ブロック1830に渡す。機能ブロック1830は、非アンカ深度ピクチャのビュー間参照依存性情報をセットし、制御を機能ブロック1835に渡す。機能ブロック1835は、構文要素pseudo_view_present_flagをセットし、制御を決定ブロック1840に渡す。決定ブロック1840は、構文要素pseudo_view_present_flagの現在値がtrueに等しいか否かを判定する。肯定の場合、制御を機能ブロック1845に渡す。さもなければ、制御を終了ブロック1899に渡す。

【0137】

機能ブロック1845は、

tiling_mode;

org_pic_width_in_mbs_minus1;及び

org_pic_height_in_mbs_minus1

の構文要素をセットし、制御を機能ブロック1850に渡す。機能ブロック1850は、符号化ビュー毎に構文要素pseudo_view_info(view_id)を呼び出し、制御を終了ブロック1899に渡す。

【0138】

図19に移れば、MPEG-4 AVC標準のマルチビュー・ビデオ符号化(MVC)の拡張を使用して複数のビュー及び深度のピクチャを符号化する例示的な方法の全体を参照符号1900で示す。

【0139】

方法1900は開始ブロック1902を含む。開始ブロック1902は、制御を機能ブロック1904に渡す。機能ブロック1904は、構文要素num_sub_views_minus1をセットし、制御を機能ブロック1906に渡す。機能ブロック1906は、変数iをゼロに等しくセットし、制御を決定ブロック1908に渡す。決定ブロック1908は、変数iがsub_viewsの数未満であるか否かを判定する。肯定の場合、制御を機能ブロック1910に渡す。さもなければ、制御を機能ブロック1920に渡す。

【0140】

機能ブロック1910は、構文要素sub_view_id[i]をセットし、制御を機能ブロック1912に渡す。機能ブロック1912は、構文要素num_parts_minus1[sub_view_id[i]]をセットし、制御を機能ブロック1914に渡す。機能ブロック1914は変数jをゼロに等しくセットし、制御を決定ブロック1916に渡す。決定ブロック1916は、変数jが構文要素num_parts_minus1[sub_view_id[i]]未満であるか否かを判定する。肯定の場合、制御を機能ブロック1918に渡す。さもなければ、制御を決定ブロック1922に渡す

。

【0141】

機能ブロック1918は、

```
loc__left__offset[sub__view__id[i]][j];
loc__top__offset[sub__view__id[i]][j];
frame__crop__left__offset[sub__view__id[i]][j];
frame__crop__right__offset[sub__view__id[i]][j];
frame__crop__top__offset[sub__view__id[i]][j];
```

10

及び

```
frame__crop__bottom__offset[sub__view__id[i]][j]
```

の構文要素をセットし、変数jを増やし、制御を決定ブロック1916に戻す。

【0142】

機能ブロック1920は、マルチビュー・ビデオ符号化(MVC)を使用して現在のビューの現在の深度を符号化し、制御を終了ブロック1999に渡す。深度信号は、その対応するビデオ信号が符号化されるやり方と同様に符号化することができる。例えば、ビューの深度信号は、他の深度信号のみ、ビデオ信号のみ、又は深度信号及びビデオ信号を含むタイル上に含まれ得る。タイル(疑似ビュー)はその場合、MVCの単一ビューとして扱われ、MVCの他のビューとして扱われる他のタイルも存在していると思われる。

20

【0143】

決定ブロック1922は、構文要素tiling__modeがゼロに等しいか否かを判定する。肯定の場合、制御を機能ブロック1924に渡す。さもなければ、制御を機能ブロック1938に渡す。

【0144】

機能ブロック1924は、構文要素flip__dir[sub__view__id[i]]及び構文要素upsample__view__flag[sub__view__id[i]]をセットし、制御を決定ブロック1926に渡す。決定ブロック1926は、構文要素upsample__view__flag[sub__view__id[i]]の現在値が1に等しいか否かを判定する。肯定の場合、制御を機能ブロック1928に渡す。さもなければ、制御を決定ブロック1930に渡す。

30

【0145】

機能ブロック1928は構文要素upsample__filter[sub__view__id[i]]をセットし、制御を決定ブロック1930に渡す。決定ブロック1930は、構文要素upsample__filter[sub__view__id[i]]の値が3に等しいか否かを判定する。肯定の場合、制御を機能ブロック1932に渡す。さもなければ、制御を機能ブロック1936に渡す。

40

【0146】

```
vert__dim[sub__view__id[i]];
hor__dim[sub__view__id[i]];及び
quantizer[sub__view__id[i]]
```

の構文要素をセットし、制御を機能ブロック1934に渡す。機能ブロック1934は、YUV成分毎にフィルタ係数をセットし、制御を機能ブロック1936に渡す。

【0147】

機能ブロック1936は変数iを増やし、制御を決定ブロック1908に渡す。

【0148】

機能ブロック1938は、構文要素pixel__dist__x[sub__view__id[i]]及び構文要素flip__dist__y[sub__view__id[i]]をセ

50

ットし、制御を機能ブロック1940に渡す。機能ブロック1940は変数jをゼロにセットし、制御を決定ブロック1942に渡す。決定ブロック1942は、変数jの現在値が構文要素 `num_parts[sub_view_id[i]]` の現在値未満であるか否かを判定する。肯定の場合、制御を機能ブロック1944に渡す。さもなければ、制御を機能ブロック1936に渡す。

【0149】

機能ブロック1944は、構文要素 `num_pixel_tiling_filter_coef fs_minus1[sub_view_id[i]]` をセットし、制御を機能ブロック1946に渡す。機能ブロック1946は、画素タイリング・フィルタ全ての係数をセットし、制御を機能ブロック1936に渡す。

10

【0150】

図20に移れば、MPEG-4 AVC標準のマルチビュー・ビデオ符号化(MVC)の拡張を使用したピクチャの復号化に備えて複数のビュー及び深度のピクチャを処理する例示的な方法の全体を参照符号2000で示す。

【0151】

方法2000は開始ブロック2005を含み、開始ブロック2005は制御を機能ブロック2015に渡す。機能ブロック2015は構文要素 `num_coded_views_minus1` を解析し、制御を機能ブロック2020に渡す。機能ブロック2020は、`view_id[i]` に対応する (`num_coded_views_minus1 + 1`) 個の深度全ての構文要素 `view_id[i]` を解析し、制御を機能ブロック2025に渡す。機能ブロック2025は、アンカ深度ピクチャのビュー間参照依存性情報を解析し、制御を機能ブロック2030に渡す。機能ブロック2030は、非アンカ深度ピクチャのビュー間参照依存性情報を解析し、制御を機能ブロック2035に渡す。機能ブロック2035は、構文要素 `pseudo_view_present_flag` を解析し、制御を決定ブロック2040に渡す。決定ブロック2040は、構文要素 `pseudo_view_present_flag` の現在値が `true` に等しいか否かを判定する。肯定の場合、制御を機能ブロック2045に渡す。さもなければ、制御を機能ブロック2099に渡す。

20

【0152】

機能ブロック2045は、
`tiling_mode` ;
`org_pic_width_in_mbs_minus1` ; 及び
`org_pic_height_in_mbs_minus1`
 の構文要素をセットし、制御を機能ブロック2050に渡す。機能ブロック2050は、符号化ビュー毎に構文要素 `pseudo_view_info(view_id)` を呼び出し、制御を終了ブロック2099に渡す。

30

【0153】

図21に移れば、MPEG-4 AVC標準のマルチビュー・ビデオ符号化(MVC)の拡張を使用して複数のビュー及び深度のピクチャを復号化する例示的な方法の全体を参照符号2100で示す。

40

【0154】

方法2100は開始ブロック2102を含む。開始ブロック2102は、入力パラメータ `pseudo_view_id` から始まり、制御を機能ブロック2104に渡す。機能ブロック2104は、構文要素 `num_sub_views_minus1` を解析し、制御を機能ブロック2106に渡す。機能ブロック2106は、変数iをゼロに等しくセットし、制御を決定ブロック2108に渡す。決定ブロック2108は、変数iが `sub_views` の数未満であるか否かを判定する。肯定の場合、制御を機能ブロック2110に渡す。さもなければ、制御を機能ブロック2120に渡す。

【0155】

機能ブロック2110は、構文要素 `sub_view_id[i]` を解析し、制御を機

50

機能ブロック2112に渡す。機能ブロック2112は、構文要素 `num_parts_minus1[sub_view_id[i]]` を解析し、制御を機能ブロック2114に渡す。機能ブロック2114は変数 `j` をゼロに等しくセットし、制御を決定ブロック2116に渡す。決定ブロック2116は、変数 `j` が構文要素 `num_parts_minus1[sub_view_id[i]]` 未満であるか否かを判定する。肯定の場合、制御を機能ブロック2118に渡す。さもなければ、制御を決定ブロック2122に渡す。

【0156】

機能ブロック2118は、

```
loc_left_offset[sub_view_id[i]][j];
loc_top_offset[sub_view_id[i]][j];
frame_crop_left_offset[sub_view_id[i]][j];
frame_crop_right_offset[sub_view_id[i]][j];
frame_crop_top_offset[sub_view_id[i]][j];
```

及び

```
frame_crop_bottom_offset[sub_view_id[i]][j]
```

の構文要素をセットし、変数 `j` を増やし、制御を決定ブロック2116に戻す。

【0157】

機能ブロック2120は、マルチビュー・ビデオ符号化(MVC)を使用して現在のピクチャを復号化し、制御を機能ブロック2121に渡す。機能ブロック2121は、各ビューをピクチャから高位構文を使用して分離し、制御を終了ブロック2199に渡す。高位構文を使用した、各ビューの分離は上述の通りである。

【0158】

決定ブロック2122は、構文要素 `tiling_mode` がゼロに等しいか否かを判定する。肯定の場合、制御を機能ブロック2124に渡す。さもなければ、制御を機能ブロック2138に渡す。

【0159】

機能ブロック2124は、構文要素 `flip_dir[sub_view_id[i]]` 及び構文要素 `upsample_view_flag[sub_view_id[i]]` を解析し、制御を決定ブロック2126に渡す。決定ブロック2126は、構文要素 `upsample_view_flag[sub_view_id[i]]` の現在値が1に等しいか否かを判定する。肯定の場合、制御を機能ブロック2128に渡す。さもなければ、制御を決定ブロック2130に渡す。

【0160】

機能ブロック2128は、構文要素 `upsample_filter[sub_view_id[i]]` を解析し、制御を決定ブロック2130に渡す。決定ブロック2130は、構文要素 `upsample_filter[sub_view_id[i]]` の値が3に等しいか否かを判定する。肯定の場合、制御を機能ブロック2132に渡す。さもなければ、制御を機能ブロック2136に渡す。

【0161】

機能ブロック2132は、

```
vert_dim[sub_view_id[i]];
hor_dim[sub_view_id[i]];及び
quantizer[sub_view_id[i]]
```

の構文要素を解析し、制御を機能ブロック2134に渡す。機能ブロック2134は、YUV成分毎にフィルタ係数を解析し、制御を機能ブロック2136に渡す。

10

20

30

40

50

【0162】

機能ブロック2136は変数*i*を増やし、制御を決定ブロック2108に戻す。

【0163】

機能ブロック2138は、構文要素 `pixel_dist_x[sub_view_id[i]]` 及び構文要素 `flip_dist_y[sub_view_id[i]]` を解析し、制御を機能ブロック2140に渡す。機能ブロック2140は変数*j*をゼロに等しくセットし、制御を決定ブロック2142に渡す。決定ブロック2142は、変数*j*の現在値が構文要素 `num_parts[sub_view_id[i]]` の現在値未満であるか否かを判定する。肯定の場合、制御を機能ブロック2144に渡す。さもなければ、制御を機能ブロック2136に渡す。

10

【0164】

機能ブロック2144は、構文要素 `num_pixel_tiling_filter_coefs_minus1[sub_view_id[i]]` を解析し、制御を機能ブロック2146に渡す。機能ブロック2146は、画素タイリング・フィルタ全ての係数を解析し、制御を機能ブロック2136に渡す。

【0165】

図22に移れば、画素レベルにおけるタイリング例の全体を参照符号2200で示す。図22は以下に更に説明する。

【0166】

MPEG-4 AVC又はMVCを使用したビューのタイリング

20

マルチビュー・ビデオ符号化のアプリケーションに、自由視点TV（すなわちFTV）がある。前述のアプリケーションは、2つ以上のビュー間でユーザが自由に移動可能であることを必要とする。このことを達成するために、2つのビュー間の「仮想」ビューを補間又は合成する必要がある。ビュー補間を実行するための手法はいくつか存在している。上記手法の1つは、ビュー補間/合成に深度を使用する。

【0167】

各ビューは、関連付けられた深度信号を有し得る。よって、深度は、ビデオ信号の別の形式とみなし得る。図9は、深度信号900の例を示す。FTVなどのアプリケーションをイネーブルするために、深度信号がビデオ信号とともに送信される。提案されたタイリング・フレームワークでは、深度信号は、タイルの1つとして加えることも可能である。図10は、タイルとして加えられた深度信号の例を示す。深度信号/タイルは、図10の右側に示す。

30

【0168】

深度が、フレーム全体のタイルとして符号化されると、高位構文は、レンダリング器が深度信号を適切に使用することが可能であるように、どのタイルが深度信号であることを示すものとする。

【0169】

（図1に示すものなどの）入力シーケンスが、MPEG-4 AVC標準符号化器（又は、別のビデオ符号化標準及び/又は勧告に対応する符号化器）を使用して符号化された場合、本願提案の高位構文は例えば、シーケンス・パラメータ・セット（SPS）、ピクチャ・パラメータ・セット（PPS）、スライス・ヘッダ、及び/又は付加拡張情報（SEI）メッセージに存在し得る。本願提案の方法の実施例を表1に示す。表1では、構文は付加拡張情報（SEI）メッセージに存在している。

40

【0170】

（図1に示すものなどの）疑似ビューの入力シーケンスが、MPEG-4 AVC標準の符号化器のマルチビュー・ビデオ符号化（MVC）の拡張（又は、別のビデオ符号化標準及び/又は勧告に関するマルチビュー・ビデオ符号化標準に対応する符号化器）を使用して符号化された場合、本願提案の高位構文がSPS、PPS、スライス・ヘッダ、SEIメッセージ、又は特定されたプロファイルに存在し得る。

【0171】

50

本願提案の方法の一実施例を表 1 に示す。表 1 は、本願の原理の実施例によって提案された構文要素を含む、シーケンス・パラメータ・セット (S P S) 構造に存在している構文要素を示す。

【 0 1 7 2 】

【 表 1 】

seq_parameter_set_mvc_extension() {	C	記述子
num_views_minus_1		ue(v)
for(i = 0; i <= num_views_minus_1; i++)		
view_id[i]		ue(v)
for(i = 0; i <= num_views_minus_1; i++) {		
num_anchor_refs_l0[i]		ue(v)
for(j = 0; j < num_anchor_refs_l0[i]; j++)		
anchor_ref_l0[i][j]		ue(v)
num_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_anchor_refs_l1[i]; j++)		
anchor_ref_l1[i][j]		ue(v)
}		
for(i = 0; i <= num_views_minus_1; i++) {		
num_non_anchor_refs_l0[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l0[i]; j++)		
non_anchor_ref_l0[i][j]		ue(v)
num_non_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l1[i]; j++)		
non_anchor_ref_l1[i][j]		ue(v)
}		
pseudo_view_present_flag		u(1)
if (pseudo_view_present_flag) {		
tiling_mode		
org_pic_width_in_mbs_minus1		
org_pic_height_in_mbs_minus1		
for(i = 0; i < num_views_minus_1; i++)		
pseudo_view_info(i);		
}		
}		

10

20

30

表 2 は、本願の原理の実施例による、表 1 の p s e u d o _ v i e w _ i n f o 構文要素の構文要素を示す。

【 0 1 7 3 】

【表 2】

	C	記述子
pseudo_view_info (pseudo_view_id) {		
num_sub_views_minus_1[pseudo_view_id]	5	ue(v)
if (num_sub_views_minus_1 != 0) {		
for (i = 0; i < num_sub_views_minus_1[pseudo_view_id]; i++) {		
sub_view_id[i]	5	ue(v)
num_parts_minus1[sub_view_id[i]]	5	ue(v)
for(j = 0; j <= num_parts_minus1[sub_view_id[i]]; j++) {		
loc_left_offset[sub_view_id[i]][j]	5	ue(v)
loc_top_offset[sub_view_id[i]][j]	5	ue(v)
frame_crop_left_offset[sub_view_id[i]][j]	5	ue(v)
frame_crop_right_offset[sub_view_id[i]][j]	5	ue(v)
frame_crop_top_offset[sub_view_id[i]][j]	5	ue(v)
frame_crop_bottom_offset[sub_view_id[i]][j]	5	ue(v)
}		
if (tiling_mode == 0) {		
flip_dir[sub_view_id[i][j]]	5	u(2)
upsample_view_flag[sub_view_id[i]]	5	u(1)
if(upsample_view_flag[sub_view_id[i]])		
upsample_filter[sub_view_id[i]]	5	u(2)
if(upsample_fiter[sub_view_id[i]] == 3) {		
vert_dim[sub_view_id[i]]	5	ue(v)
hor_dim[sub_view_id[i]]	5	ue(v)
quantizer[sub_view_id[i]]	5	ue(v)
for (yuv = 0; yuv < 3; yuv++) {		
for (y = 0; y < vert_dim[sub_view_id[i]] - 1; y++) {		
for (x = 0; x < hor_dim[sub_view_id[i]] - 1; x++)		
filter_co coeffs[sub_view_id[i]] [yuv][y][x]	5	se(v)
}		
}		
}		
}		
}		
} // if(tiling_mode == 0)		
else if (tiling_mode == 1) {		
pixel_dist_x[sub_view_id[i]]		
pixel_dist_y[sub_view_id[i]]		
for(j = 0; j <= num_parts[sub_view_id[i]]; j++) {		
num_pixel_tiling_filter_co coeffs_minus1[sub_view_id[i]][j]		
for (coeff_idx = 0; coeff_idx <=		
num_pixel_tiling_filter_co coeffs_minus1[sub_view_id[i]][j]; j++)		
pixel_tiling_filter_co coeffs[sub_view_id[i]][j]		
} // for (j = 0; j <= num_parts[sub_view_id[i]]; j++)		
} // else if (tiling_mode == 1)		
} // for (i = 0; i < num_sub_views_minus_1; i++)		
} // if (num_sub_views_minus_1 != 0)		
}		

10

20

30

40

表 1 及び表 2 に提示された構文要素の意味規則

pseudo_view_present_flag が true に等しいことは、特定のビューが、複数のサブビューのスーパービューであることを示している。

【 0 1 7 4 】

tiling_mode が 0 に等しいことは、サブビューがピクチャ・レベルでタイリ

50

ングされる旨を示す。1の値は、タイリングが画素レベルで行われることを示す。

【0175】

新たなSEIメッセージは、MPEG-4 AVC標準、又はMPEG-4 AVC標準の拡張において使用されていないSEIペイロード・タイプの値を使用することが可能である。新たなSEIメッセージは、以下の意味規則を有するいくつかの構文要素を含む。

【0176】

$num_coded_views_minus\ 1\ plus\ 1$ は、ビットストリームによってサポートされる符号化ビューの数を示す。 $num_coded_views_minus\ 1$ の値は、0以上1023以下の範囲内にある。

【0177】

$org_pic_width_in_mbs_minus\ 1\ plus\ 1$ は、各ビューにおけるピクチャの幅をマクロブロック単位で規定する。

マクロブロック単位での、ピクチャ幅の変数は以下のように導き出される。

【0178】

$PicWidthInMbs = org_pic_width_in_mbs_minus\ 1 + 1$

luma成分のピクチャ幅の変数は以下のように導き出される。

【0179】

$PicWidthInSamplesL = PicWidthInMbs * 16$

chroma成分のピクチャ幅の変数は以下のように導き出される。

【0180】

$PicWidthInSamplesC = PicWidthInMbs * MbWidthC$

$org_pic_height_in_mbs_minus\ 1\ plus\ 1$ は、各ビューにおけるピクチャの高さをマクロブロック単位で規定する。

マクロブロック単位での、ピクチャの高さの変数は以下のように導き出される。

【0181】

$PicHeightInMbs = org_pic_height_in_mbs_minus\ 1 + 1$

luma成分のピクチャの高さの変数は以下のように導き出される。

【0182】

$PicHeightInSamplesL = PicHeightInMbs * 16$

chroma成分のピクチャの高さの変数は以下のように導き出される。

【0183】

$PicHeightInSamplesC = PicHeightInMbs * MbHeightC$

$num_sub_views_minus\ 1\ plus\ 1$ は、現在のビューに含まれる符号化されたサブビューの数を示す。 $num_coded_views_minus\ 1$ の値は、0以上1023以下の範囲内にある。

【0184】

$sub_view_id[i]$ は、 i で示す復号化順序のサブビューの sub_view_id を規定する。

【0185】

$num_parts[sub_view_id[i]]$ は、 $sub_view_id[i]$ のピクチャが分割される部分の数を規定する。

【0186】

$loc_left_offset[sub_view_id[i]][j]$ 及び $loc_top_offset[sub_view_id[i]][j]$ はそれぞれ、左画素オフセット及び上画素オフセットにおける位置を規定し、現在の部分 j は、ビューの最終再

10

20

30

40

50

構成ピクチャ内に配置されており、`sub_view_id`は`sub_view_id[i]`に等しい。

【0187】

`view_id[i]`は、ビューの`view_id`を規定し、符号化順序は`i`で示す。

【0188】

`frame_crop_left_offset[view_id[i]][j]`、`frame_crop_right_offset[view_id[i]][j]`、`frame_crop_top_offset[view_id[i]][j]`及び`frame_crop_bottom_offset[view_id[i]][j]`は、出力するためにフレーム座標に規定された矩形領域について、`num_part_j`及び`view_id_i`の一部である符号化ビデオ・シーケンスにおけるピクチャのサンプルを規定する。

10

【0189】

変数`CropUnitX`及び`CropUnitY`は以下のように導き出される。

【0190】

`chroma_format_idc`が0に等しい場合、`CropUnitX`及び`CropUnitY`は以下のように導き出される。

【0191】

$CropUnitX = 1$

$CropUnitY = 2 - frame_mbs_only_flag$

20

さもなければ(`chroma_format_idc`は、1、2又は3に等しい場合)

`CropUnitX`及び`CropUnitY`は以下のように導き出される。

【0192】

$CropUnitX = SubWidthC$

$CropUnitY = SubHeightC * (2 - frame_mbs_only_flag)$

フレーム・クロップ矩形は、

$CropUnitX * frame_crop_left_offset$ 乃至 $PicWidthInSamplesL - (CropUnitX * frame_crop_right_offset + 1)$ の水平フレーム座標を備えた`luma`サンプルを含む。垂直フレーム座標は、 $CropUnitY * frame_crop_top_offset$ 以上($16 * FrameHeightInMbs$) - ($CropUnitY * frame_crop_bottom_offset + 1$)以下である。`frame_crop_left_offset`の値は、0以上($PicWidthInSamplesL / CropUnitX$) - ($frame_crop_right_offset + 1$)以下の範囲内にあるものとし、`frame_crop_top_offset`の値は、0以上($16 * FrameHeightInMbs / CropUnitY$) - ($frame_crop_bottom_offset + 1$)以下の範囲内にあるものとする。

30

40

【0193】

`chroma_format_idc`が0に等しくない場合、2つの`chroma`アレイの対応する規定サンプルは、フレーム座標($x / SubWidthC$, $y / SubHeightC$)を有するサンプルであり、(x , y)は規定`luma`サンプルのフレーム座標である。

【0194】

復号化フィールドの場合、復号化フィールドの規定されたサンプルは、フレーム座標に規定された矩形内に収まるサンプルである。

【0195】

`num_parts[view_id[i]]`は、`view_id[i]`のピクチャが

50

分割される部分の数を規定する。

【0196】

`depth_flag[view_id[i]]`は、現在の部分が深度信号であるか否かを規定する。`depth_flag`が0に等しい場合、現在の部分は深度信号でない。`depth_flag`が1に等しい場合、現在の部分は、`view_id[i]`で識別されるビューに関連付けられた深度信号である。

【0197】

`flip_dir[sub_view_id[i]][j]`は、現在の部分の反転方向を規定する。`flip_dir`が0に等しいことは反転がないことを意味し、`flip_dir`が1に等しいことは水平方向の反転を示し、`flip_dir`が2に等しいことは垂直方向の反転を示し、`flip_dir`が3に等しいことは水平方向及び垂直方向の反転を示す。

10

【0198】

`flip_dir[view_id[i]][j]`は現在の部分の反転方向を規定し、`flip_dir`が0に等しいことは反転がない状態を示し、`flip_dir`が1に等しいことは水平方向における反転を示し、`flip_dir`が2に等しいことは垂直方向における反転を示し、`flip_dir`が3に等しいことは水平方向及び垂直方向における反転を示す。

【0199】

`loc_left_offset[view_id[i]][j]`及び`loc_top_offset[view_id[i]][j]`は、画素オフセットにおける位置を規定し、現在の部分jは、ビューの最終再構成ピクチャ内に配置されており、`view_id`は`id[i]`に等しい。

20

【0200】

`upsample_view_flag[view_id[i]]`は、`view_id[i]`で規定されるビューに属するピクチャをアップサンプリングする必要があるか否かを示す。`upsample_view_flag[view_id[i]]`が0に等しいことは、`view_id`が`view_id[i]`に等しいピクチャがアップサンプリングされないことを規定し、`upsample_view_flag[view_id[i]]`が1に等しいことは、`view_id`が`view_id[i]`に等しいピクチャがアップサンプリングされることを規定している。

30

【0201】

`upsample_filter[view_id[i]]`はアップサンプリングに使用すべきフィルタのタイプを示す。`upsample_filter[view_id[i]]`が0に等しいことは、6タップのAVCフィルタを使用すべきであることを示し、`upsample_filter[view_id[i]]`が1に等しいことは、4タップのSVCフィルタを使用すべきであることを示し、`upsample_filter[view_id[i]]`が2であることは、バイリニア・フィルタを使用すべきであることを示し、`upsample_filter[view_id[i]]`が3であることは、カスタム・フィルタ係数が送信される旨を示す。`upsample_fiter[view_id[i]]`は、存在しない場合、0にセットされる。この実施例では、カスタム化された2Dフィルタを使用する。これは、1Dフィルタに、及び特定の他の非線形フィルタに容易に拡張することが可能である。

40

【0202】

`vert_dim[view_id[i]]`は、カスタムの2Dフィルタの垂直方向の寸法を規定する。

【0203】

`hor_dim[view_id[i]]`は、カスタムの2Dフィルタの水平方向の寸法を規定する。

【0204】

50

`quantizer[view_id[i]]`は、各フィルタ係数の量子化因子を規定する。

【0205】

`filter_coeffs[view_id[i]][yuv][y][x]`は量子化フィルタ係数を規定し、`yuv`は、フィルタ係数が該当する成分を表し、`yuv`が0に等しいことはY成分を規定し、`yuv`が1に等しいことはU成分を規定し、`yuv`が2に等しいことはV成分を規定する。

【0206】

`pixel_dist_x[sub_view_id[i]]`及び`pixel_dist_y[sub_view_id[i]]`はそれぞれ、ビュー内の近傍画素間の最終的な再構成疑似ビュー内の水平方向及び垂直方向における距離を規定する。`sub_view_id`は`sub_view_id[i]`に等しい。

10

【0207】

`num_pixel_tiling_filter_coeffs_minus1[sub_view_id[i]][j] plus one`は、タイリング・モードが1に等しくセットとされると、フィルタ係数の数を示す。

【0208】

`pixel_tiling_filter_coeffs[sub_view_id[i]][j]`は、タイリングされたピクチャをフィルタリングするよう使用することができるフィルタを表すために必要なフィルタ係数を表す。

20

【0209】

画素レベルにおけるタイリング例

図22に移れば、4つのビューからの画素をタイリングすることによる疑似ビューの合成を示す2つの例はそれぞれ、参照符号2210及び2220それぞれで示す。4つのビューは併せて、参照符号2250で示す。図22における第1の例の構文値は以下、表3に記載する。

【0210】

【表 3】

pseudo_view_info (pseudo_view_id) {	値
num_sub_views_minus_1[pseudo_view_id]	3
sub_view_id[0]	0
num_parts_minus1[0]	0
loc_left_offset[0][0]	0
loc_top_offset[0][0]	0
pixel_dist_x[0][0]	0
pixel_dist_y[0][0]	0
sub_view_id[1]	0
num_parts_minus1[1]	0
loc_left_offset[1][0]	1
loc_top_offset[1][0]	0
pixel_dist_x[1][0]	0
pixel_dist_y[1][0]	0
sub_view_id[2]	0
num_parts_minus1[2]	0
loc_left_offset[2][0]	0
loc_top_offset[2][0]	1
pixel_dist_x[2][0]	0
pixel_dist_y[2][0]	0
sub_view_id[3]	0
num_parts_minus1[3]	0
loc_left_offset[3][0]	1
loc_top_offset[3][0]	1
pixel_dist_x[3][0]	0
pixel_dist_y[3][0]	0

10

20

30

40

50

図 2 2 における第 2 の例の構文値は、2 つの構文要素 (loc _ l e f t _ o f f s e t [3] [0] が 5 に等しく、 loc _ t o p _ o f f s e t [3] [0] が 3 に等しい) 以外は全て、同じである。

【 0 2 1 1 】

ビューに対応する画素が、特定のオフセット位置で開始すべきである旨をオフセットは示す。このことは図 2 2 (2 2 2 0) に示す。このことは例えば、共通の物体がビュー間でシフトしている画像を 2 つのビューが生成する場合に行うことができる。例えば、(第 1 のビュー及び第 2 のビューを表す) 第 1 のカメラ及び第 2 のカメラが物体の写真を撮影した場合、物体は、第 1 のビューと比較して、第 2 のビューにおける右に 5 個の画素だけ、移動させられているように見え得る。第 1 のビューにおける画素 (i - 5 , j) は第 2 のビューにおける画素 (i , j) に対応することを意味している。2 つのビューの画素が、画素単位でタイリングされているに過ぎない場合、タイル内の近傍画素間であまり相関がないことがあり得、特殊な符号化利得は小さいことがあり得る。逆に、ビュー 1 からの画素 (i - 5 , j) がビュー 2 からの画素 (i , j) の隣に配置されるようにタイリングをシフトさせることにより、空間的相関を増加させることができる、空間符号化利得も増加させることができる。例えば、第 1 のビュー及び第 2 のビューにおける物体の対応する画素が隣同士でタイリングされているので、このことが当てはまる。

【 0 2 1 2 】

よって、 loc _ l e f t _ o f f s e t 及び loc _ t o p _ o f f s e t の存在は符号化効率を向上させ得る。オフセット情報は外部手段によって得ることができる。例えば、カメラの位置情報又はビュー間の大局的な視差ベクトルを使用して前述のオフセット情報を求めることができる。

【0213】

オフセット化の結果、疑似ビューにおける画素の一部には、何れのビューからの画素値にも割り当てられない。上記例を続ければ、ビュー2からの画素(i, j)と並んで、ビュー1からの画素($i-5, j$)をタイリングする場合、 $i = 0 \dots 4$ の値の場合、タイル対象のビュー1からの画素($i-5, j$)は存在しないので、前述の画素はタイルにおいて空き状態である。何れのビューからの画素値も割り当てられない、疑似ビュー(タイル)における画素について、少なくとも1つの実現形態は、AVCにおける動き補償のサブ画素補間手順と同様な補間手順を使用する。すなわち、空き状態のタイル画素は近傍画素から補間することができる。前述の補間は、タイルにおける、より大きな空間的相関、及びタイルのより大きな符号化利得をもたらし得る。

10

【0214】

ビデオ符号化では、Iピクチャ、PピクチャやBピクチャなどのピクチャ毎に別々の符号化タイプを選ぶことが可能である。更に、マルチビュー・ビデオ符号化の場合、アンカ・ピクチャ及び非アンカ・ピクチャを定義する。一実施例では、グループ化の決定をピクチャ・タイプに基づいて行うことが可能である旨を提案する。前述のグループ化の情報は高位構文においてシグナリングされる。

【0215】

図11に移れば、単一のフレーム上にタイリングされた5つのビューの例の全体を参照符号1100で示す。特に、単一のフレーム上にタイリングされた5つのビューを備えた、ボールルームのシーケンスを示す。更に、5番目のビューは、矩形のフレーム上に配置することが可能であるように2つの部分に分割されている。ここでは、各ビューは、合計フレーム寸法が 640×640 であるようなQVGAサイズのものである。600は、16の倍数でないので、608に増やすものとする。

20

【0216】

この例の場合、考えられるSEIメッセージは表4に示すようなものであり得る。

【0217】

【表 4 - 1】

multiview_display_info(payloadSize) {	値
num_coded_views_minus1	5
org_pic_width_in_mbs_minus1	40
org_pic_height_in_mbs_minus1	30
view_id[0]	0
num_parts[view_id[0]]	1
depth_flag[view_id[0]][0]	0
flip_dir[view_id[0]][0]	0
loc_left_offset[view_id[0]][0]	0
loc_top_offset[view_id[0]][0]	0
frame_crop_left_offset[view_id[0]][0]	0
frame_crop_right_offset[view_id[0]][0]	320
frame_crop_top_offset[view_id[0]][0]	0
frame_crop_bottom_offset[view_id[0]][0]	240
upsample_view_flag[view_id[0]]	1
if(upsample_view_flag[view_id[0]]) {	
vert_dim[view_id[0]]	6
hor_dim[view_id[0]]	6
quantizer[view_id[0]]	32
for (yuv= 0; yuv< 3; yuv++) {	
for (y = 0; y < vert_dim[view_id[i]] - 1; y ++) {	
for (x = 0; x < hor_dim[view_id[i]] - 1; x ++)	
filter_coefs[view_id[i]] [yuv][y][x]	XX
view_id[1]	1
num_parts[view_id[1]]	1
depth_flag[view_id[0]][0]	0
flip_dir[view_id[1]][0]	0
loc_left_offset[view_id[1]][0]	0
loc_top_offset[view_id[1]][0]	0
frame_crop_left_offset[view_id[1]][0]	320
frame_crop_right_offset[view_id[1]][0]	640
frame_crop_top_offset[view_id[1]][0]	0
frame_crop_bottom_offset[view_id[1]][0]	320
upsample_view_flag[view_id[1]]	1
if(upsample_view_flag[view_id[1]]) {	
vert_dim[view_id[1]]	6
hor_dim[view_id[1]]	6
quantizer[view_id[1]]	32
for (yuv= 0; yuv< 3; yuv++) {	
for (y = 0; y < vert_dim[view_id[i]] - 1; y ++) {	
for (x = 0; x < hor_dim[view_id[i]] - 1; x ++)	
filter_coefs[view_id[i]] [yuv][y][x]	XX
.....(similarly for view 2,3)	
view_id[4]	4
num_parts[view_id[4]]	2
depth_flag[view_id[0]][0]	0
flip_dir[view_id[4]][0]	0

10

20

30

40

【表 4 - 2】

loc_left_offset[view_id[4]][0]	0
loc_top_offset[view_id[4]][0]	0
frame_crop_left_offset[view_id[4]][0]	0
frame_crop_right_offset[view_id[4]][0]	320
frame_crop_top_offset[view_id[4]][0]	480
frame_crop_bottom_offset[view_id[4]][0]	600
flip_dir[view_id[4]][1]	0
loc_left_offset[view_id[4]][1]	0
loc_top_offset[view_id[4]][1]	120
frame_crop_left_offset[view_id[4]][1]	320
frame_crop_right_offset[view_id[4]][1]	640
frame_crop_top_offset[view_id[4]][1]	480
frame_crop_bottom_offset[view_id[4]][1]	600
upsample_view_flag[view_id[4]]	1
if(upsample_view_flag[view_id[4]]) {	
vert_dim[view_id[4]]	6
hor_dim[view_id[4]]	6
quantizer[view_id[4]]	32
for (yuv= 0; yuv< 3; yuv++) {	
for (y = 0; y < vert_dim[view_id[4]] - 1; y ++)	
for (x = 0; x < hor_dim[view_id[4]] - 1; x ++)	
filter_coefs[view_id[4]][yuv][y][x]	XX
}	
}	

10

20

表 5 は、表 4 に示す例のマルチビュー情報を送信するための一般的な構文構造を示す。

【 0 2 1 8 】

【表 5】

	C	記述子
multiview_display_info(payloadSize) {		
num_coded_views_minus1	5	ue(v)
org_pic_width_in_mbs_minus1	5	ue(v)
org_pic_height_in_mbs_minus1	5	ue(v)
for(i = 0; i <= num_coded_views_minus1; i++) {		
view_id[i]	5	ue(v)
num_parts[view_id[i]]	5	ue(v)
for(j = 0; j <= num_parts[i]; j++) {		
depth_flag[view_id[i]][j]		
flip_dir[view_id[i]][j]	5	u(2)
loc_left_offset[view_id[i]][j]	5	ue(v)
loc_top_offset[view_id[i]][j]	5	ue(v)
frame_crop_left_offset[view_id[i]][j]	5	ue(v)
frame_crop_bottom_offset[view_id[i]][j]	5	ue(v)
}		
upsample_view_flag[view_id[i]]	5	u(1)
if(upsample_view_flag[view_id[i]])		
upsample_filter[view_id[i]]	5	u(2)
if(upsample_fiter[view_id[i]] == 3) {		
vert_dim[view_id[i]]	5	ue(v)
hor_dim[view_id[i]]	5	ue(v)
quantizer[view_id[i]]	5	ue(v)
for (yuv= 0; yuv< 3; yuv++) {		
for (y = 0; y < vert_dim[view_id[i]] - 1; y ++) {		
for (x = 0; x < hor_dim[view_id[i]] - 1; x ++)		
filter_coefs[view_id[i]] [yuv][y][x]	5	se(v)
}		
}		
}		
}		
}		
}		

10

20

30

図 2 3 を参照すれば、ビデオ処理装置 2 3 0 0 を示す。ビデオ処理装置 2 3 0 0 は例えば、符号化ビデオを受け取るセトトップ・ボックスや他の装置であり得、例えば、ユーザに向けて表示するために、又は記憶するために、復号化ビデオを供給する。よって、装置 2 3 0 0 はその出力をテレビ受像機、コンピュータ・モニタ、又はコンピュータ若しくは他の処理装置に供給することができる。

【 0 2 1 9 】

装置 2 3 0 0 は、データ信号 2 3 2 0 を受信する復号化器 2 3 1 0 を含む。データ信号 2 3 2 0 は例えば、A V C 又は M V C 互換ストリームを含み得る。復号化器 2 3 1 0 は、受信信号 2 3 2 0 の全部又は一部を復号化し、復号化ビデオ信号 2 3 3 0 及びタイリング情報 2 3 4 0 を出力として供給する。復号化ビデオ 2 3 3 0 及びタイリング情報 2 3 4 0 は選択器 2 3 5 0 に供給される。装置 2 3 0 0 は、ユーザ入力 2 3 7 0 を受信するユーザ・インタフェースも含む。ユーザ・インタフェース 2 3 6 0 は、ユーザ入力 2 3 7 0 に基づいてピクチャ選択信号 2 3 8 0 を選択器 2 3 5 0 に供給する。ピクチャ選択信号 2 3 8 0 及びユーザ入力 2 3 7 0 は、ユーザが表示させたいピクチャが複数のピクチャのうちのどれであることを示す。選択器 2 3 5 0 は、選択されたピクチャを出力 2 3 9 0 として供給する。選択器 2 3 5 0 は、ピクチャ選択情報 2 3 8 0 を使用して、出力 2 3 9 0 として、復号化ビデオ 2 3 3 0 におけるピクチャのうちのどれを供給するかを選択する。選択器 2

40

50

350は、タイリング情報2340を使用して、復号化ビデオ2330において選択ピクチャを位置特定する。

【0220】

種々の実現形態では、選択器2350はユーザ・インタフェース2360を含み、他の実現形態では、選択器2350が、別個のインタフェース機能が実行されることなく、ユーザ入力2370を直接受け取るので、ユーザ・インタフェース2360は必要でない。選択器2350は、例えば、ソフトウェアで、又は集積回路として実現することができる。選択器2350は、復号化器2310も組み入れることができる。

【0221】

より一般的には、本出願記載の種々の実現形態の復号化器は、タイル全体を含む復号化出力を供給することができる。更に、又はあるいは、復号化器は、タイルからの1つ又は複数の選択されたピクチャ（例えば、画像又は深度信号）のみを含む復号化出力を供給することができる。

10

【0222】

前述の通り、高位構文を使用して、本願の原理の1つ又は複数の実施例によるシグナリングを実行することができる。例えば、これらに限定されるものでないが、以下の何れかをシグナリングするために高位構文を使用することができる。より大きなフレームに存在している符号化ビューの数、ビュー全ての元の幅及び高さ、符号化ビュー毎の、ビューに対応するビュー識別子、符号化ビュー毎の、ビューのフレームが分割される部分の数、ビューの部分毎の反転方向（例えば、反転なし、水平方向の反転のみ、垂直方向の反転のみ、又は水平方向及び垂直方向の反転）、ビューの部分毎の、ビューの最終フレームにおいて現在の部分が属する左の位置（画素又はマクロブロックの数）、ビューの部分毎の、ビューの最終フレームにおいて現在の部分が属する部分の上の位置（画素又はマクロブロックの数）、ビューの部分毎の、クロッピング・ウィンドウの現在の大きな復号化/符号化フレームにおける左の位置（画素又はマクロブロックの数）、ビューの部分毎の、クロッピング・ウィンドウの現在の大きな復号化/符号化フレームにおける右の位置（画素又はマクロブロックの数）、ビューの部分毎の、クロッピング・ウィンドウの現在の大きな復号化/符号化フレームにおける上の位置（画素又はマクロブロックの数）、及び、ビューの部分毎の、クロッピング・ウィンドウの現在の大きな復号化/符号化フレームにおける下の位置（画素又はマクロブロックの数）、符号化ビュー毎に、出力前にビューをアップサンプリングする必要があるか否か（アップサンプリングを行う必要がある場合、高位構文を使用して、アップサンプリング方法（これらに限定されないが、6タップのAVCフィルタ、4タップのSVCフィルタ、バイリニア・フィルタ、又はカスタムの1D、2D線形若しくは非線形フィルタ）を示すことができる。

20

30

【0223】

「符号化器」及び「復号化器」の語は、一般的な構造を意味し、何れの特定の機能又は構成にも限定されない。例えば、復号化器は、符号化ビットストリームを収容する変調搬送波を受け取り、符号化ビットストリームを復調し、ビットストリームを復号化することができる。

【0224】

種々の方法を説明している。前述の方法の多くは、十分な開示をもたらすよう詳説している。しかし、前述の方法について説明された特定の構成の1つ又は多くを変えることができる変形が考えられる。更に、記載された構成の多くは当技術分野において知られており、よって、詳細には説明しないものとする。

40

【0225】

更に、いくつかの実現形態において、特定の情報を送出手のために高位構文を使用することを参照している。しかし、他の実現形態は、低位構文を使用するか、又は実際には、（例えば、符号化データの一部としての情報の送出などの）全く別の機構を使用して、同じ情報（又はその情報の変形）を供給する。

【0226】

50

種々の実現形態は、複数のビュー（一般的には、ピクチャ）を、単一のピクチャにタイリングし、単一のピクチャとして符号化し、単一のピクチャとして送出することを可能にするためにタイリング及び適切なシグナリングを提供する。シグナリング情報は、ポストプロセッサがビュー/ピクチャを引き離すことを可能にし得る。更に、タイリングされる複数のピクチャはビューであり得るが、ピクチャのうち少なくとも1つは深度情報であり得る。前述の実現形態は1つ又は複数の利点をもたらす。例えば、ユーザは、タイリングされたかたちで複数のビューを表示したいことがあり得、前述の種々の実現形態は、タイリングされたかたちで符号化し、送信/記憶する前にタイリングすることにより、前述のビューを符号化し、送信又は記憶する効率的なやり方をもたらす。

【0227】

A V C及び/又はM V Cのコンテキストで複数のビューをタイリングする実現形態も更なる利点をもたらす。A V Cは、表面的には、単一のビューのみに使用されるので、更なるビューは期待されない。しかし、前述のA V Cベースの実現形態は、A V C環境において複数のビューをもたらすことが可能である。タイリングされたピクチャが別々のビューに属する（例えば、疑似ビューにおける左右ピクチャがビュー1であり、右上ピクチャがビュー2である等）旨を復号化器が分かるように、タイリングされたビューを配置することが可能であるからである。

【0228】

更に、M V Cは既に複数のビューを含むので、複数のビューが、単一の疑似ビューに含まれることは期待されない。更に、M V Cは、サポートすることが可能なビューの数に限度を有し、前述のM V Cベースの実現形態により、（A V Cベースの実現形態におけるように）更なるビューをタイリングすることを可能にすることによってサポートすることが可能なビューの数が事実上増える。例えば、各疑似ビューは、M V Cのサポートされたビューのうち1つに対応し得、復号化器は、「サポートされたビュー」それぞれが、予め配置されたタイリング順序における4つのビューを実際を含むことが分かり得る。よって、前述の実現形態では、考えられるビューの数は、「サポートされたビュー」の数の4倍である。

【0229】

本明細書及び特許請求の範囲記載の実現形態は、例えば、方法若しくは処理、装置、又はソフトウェア・プログラムで実現することができる。単一の形式の実現形態のコンテキストでしか（例えば、方法としてしか）記載されていない場合でも、記載された特徴の実現形態は、他の形態（例えば、装置又はプログラム）で実現することも可能である。装置は、例えば、適切なハードウェア、ソフトウェア、及びファームウェアで実現することができる。方法は例えば、コンピュータ、マイクロプロセッサ、集積回路や、プログラマブル論理装置を含む、一般に処理装置を表す装置（例えば、プロセッサ）で実現することができる。処理装置は、エンドユーザ間の情報の通信を容易にする通信装置（例えば、コンピュータ、携帯電話機、ポータブル/パーソナル携帯情報端末（「PDA」）や他の装置）も含む。

【0230】

本明細書及び特許請求の範囲記載の種々の処理及び構成の実現形態は、各種の機器又はアプリケーション（特に、例えば、データ符号化及び復号化に関連付けられた機器又はアプリケーション）において実施することができる。機器の例には、ビデオ符号化器、ビデオ復号化器、ビデオ・コーデック、ウェブ・サーバ、セットトップ・ボックス、ラップトップ、パソコン、携帯電話機、PDAや、他の通信装置が含まれる。機器は明らかに、モバイルであり得、モバイル車両に設置されていてもよい。

【0231】

更に、方法は、プロセッサによって行われる命令によって実現することができ、前述の命令は、プロセッサ読み取り可能な媒体（例えば、集積回路、ソフトウェア担体や他の記憶装置（例えば、ハード・ディスク、コンパクト・ディスク、ランダム・アクセス・メモリ（「RAM」）やリードオンリー・メモリ（「ROM」）））上に記憶することが

10

20

30

40

50

できる。命令は、プロセッサ読み取り可能な媒体上に有形に実施されたアプリケーション・プログラムを構成することができる。プロセッサは明らかに、例えば、処理を行うための命令を有するプロセッサ読み取り可能な媒体を含み得る。前述のアプリケーション・プログラムは、何れかの適切なアーキテクチャを有するマシンにアップロードし、前述のマシンによって実行することができる。好ましくは、マシンは、1つ又は複数の中央処理装置(「CPU」)、ランダム・アクセス・メモリ(「RAM」)、及び入出力(「I/O」)のインタ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に実現される。コンピュータ・プラットフォームは、オペレーティング・システム及びマイクロ命令コードも含み得る。本明細書及び特許請求の範囲記載の種々の処理及び機能は、CPUによって実行することができるアプリケーション・プログラムの一部若しくはマイクロ命令コードの一部(又はそれらの組み合わせ)であり得る。更に、種々の他の周辺装置を、更なるデータ記憶装置や、印刷装置などのコンピュータ・プラットフォームに接続することができる。

10

20

30

40

【0232】

当業者に分かるように、実現形態は、例えば、記憶又は伝送することができる情報を搬送するようフォーマットされた信号を生成することも可能である。情報は例えば、前述の実現形態のうちの一つによって生成されるデータ、又は方法を行うための命令を含み得る。前述の信号は例えば、電磁波(例えば、スペクトルの無線周波数部分を使用して)として、又はベースバンド信号としてフォーマットすることができる。フォーマットは例えば、データ・ストリームの符号化、構文の生成、及び、符号化データ・ストリーム及び構文による搬送波の変調を含み得る。信号が搬送する情報は例えば、アナログ情報又はデジタル情報であり得る。信号は、知られているように、各種有線リンク又は無線リンクを介して伝送することができる。

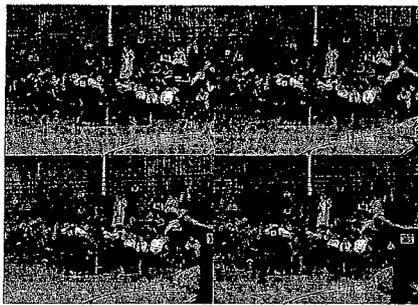
【0233】

更に、添付図面に表す構成システム部分及び方法の一部は好ましくはソフトウェアで実現されるので、システム構成部分(又は処理機能ブロック)間の実際の接続は、本願の原理がプログラムされるやり方によって変わり得る。本明細書及び特許請求の範囲記載の教示があれば、当業者は、本願の原理の前述及び同様な実現形態又は構成に想到することができるであろう。

【0234】

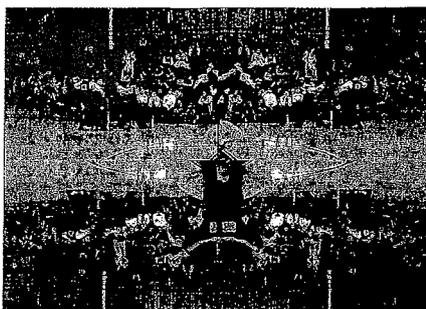
いくつかの実現形態を説明してきた。しかし、種々の修正を行うことができることが理解されるであろう。例えば、別々の実現形態の構成要素を合成、補完、修正、又は除外して他の実現形態をもたらすことができる。更に、他の構造及び処理を、開示されたものと置き換えることができ、結果として生じる実現形態は少なくとも実質的に同じ機能を少なくとも実質的に同じやり方で行って、開示された実現形態と、少なくとも実質的に同じ結果を達成することを当業者は理解するであろう。特に、例証的な実施例を添付図面を参照して本明細書及び特許請求の範囲において説明してきたが、本願の原理は前述のまさにその実施例に限定されず、本願の原理の範囲又は趣旨から逸脱することなく、種々の変更及び修正を当業者によって行うことができる。よって、前述及び他の実現形態は、本出願によって考察することができ、特許請求の範囲記載の範囲内に収まる。

【図 1】



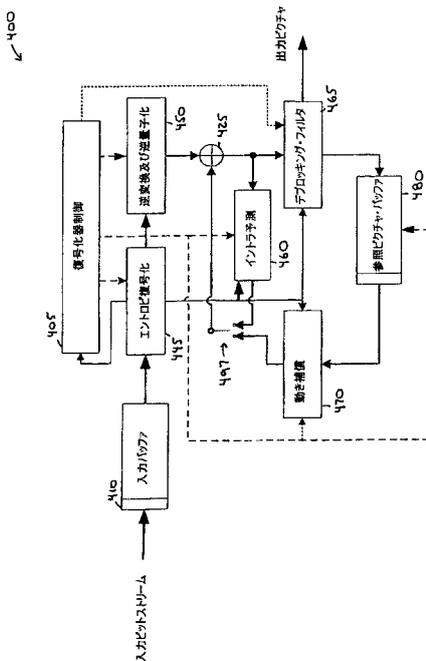
100

【図 2】

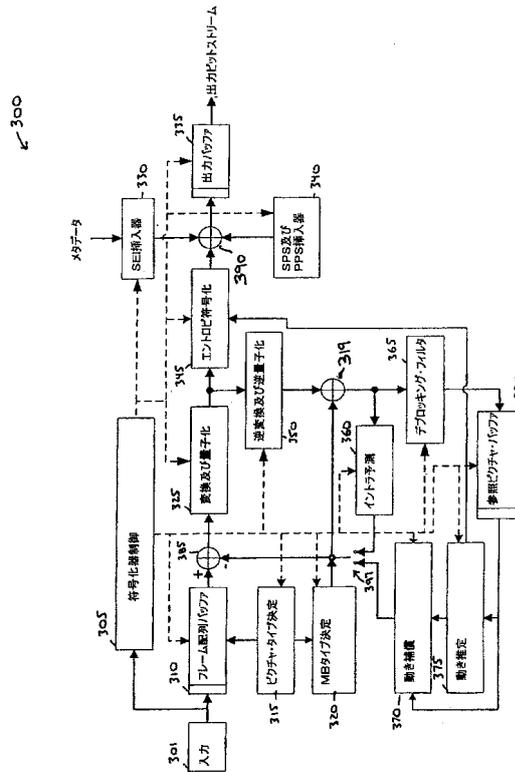


200

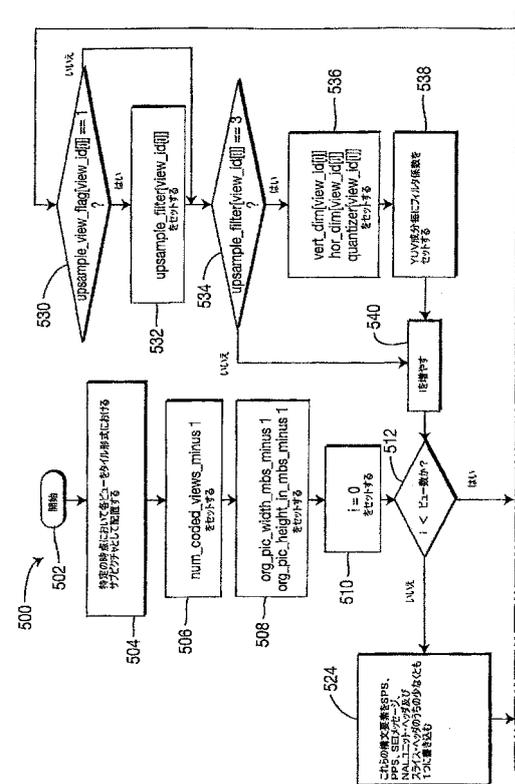
【図 4】



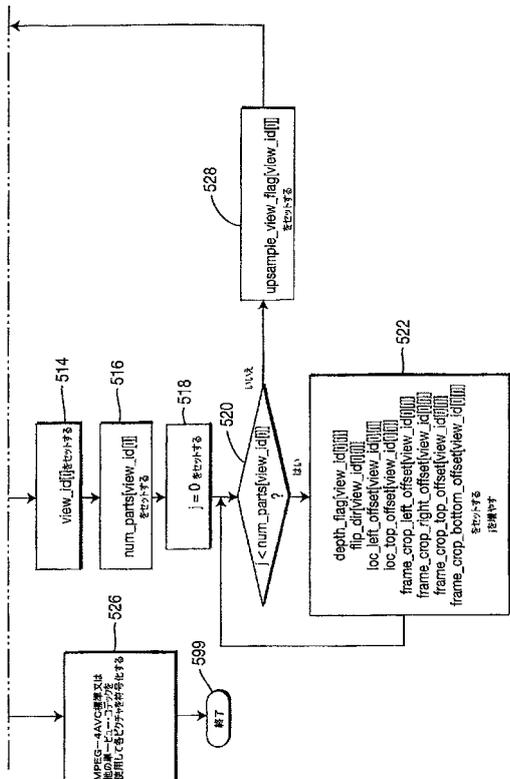
【図 3】



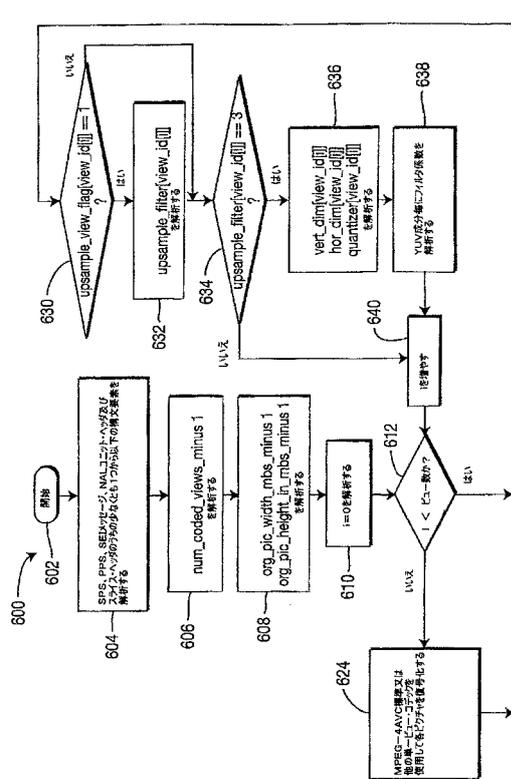
【図 5 A】



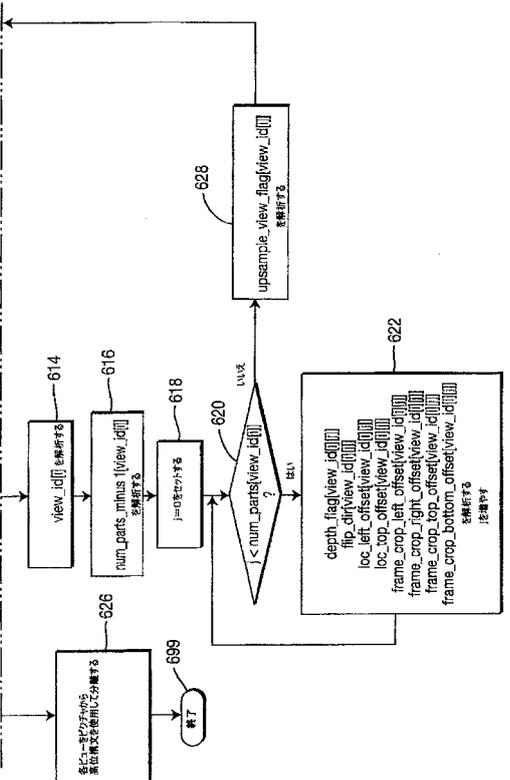
【図 5 B】



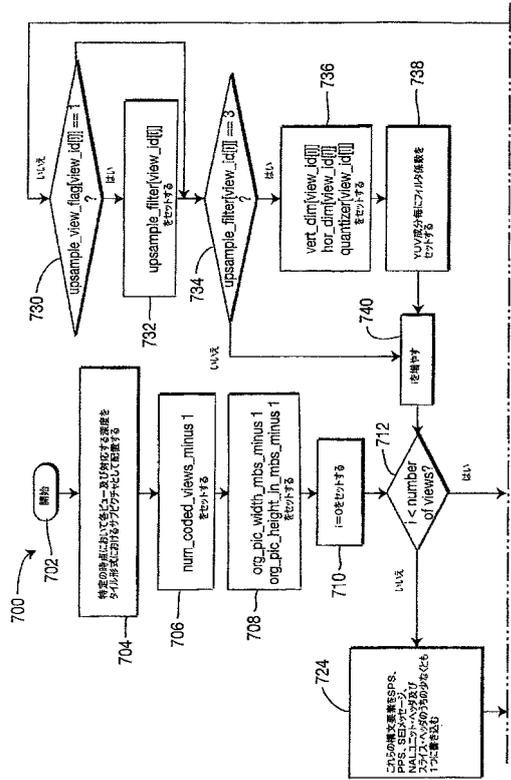
【図 6 A】



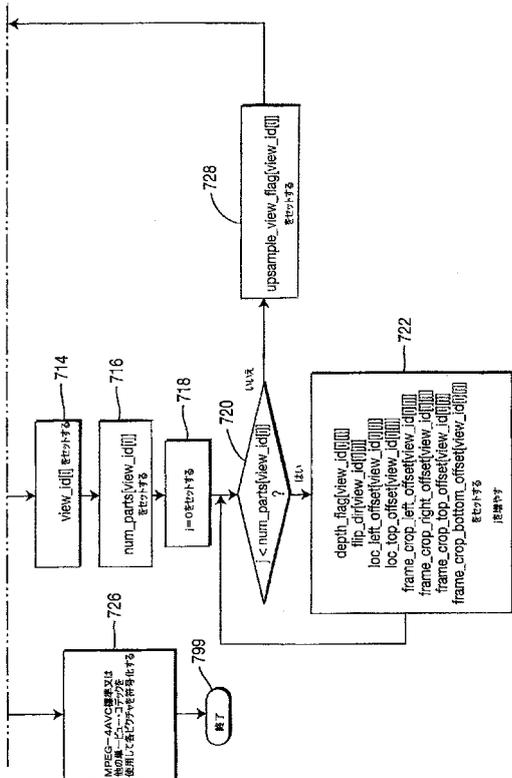
【図 6 B】



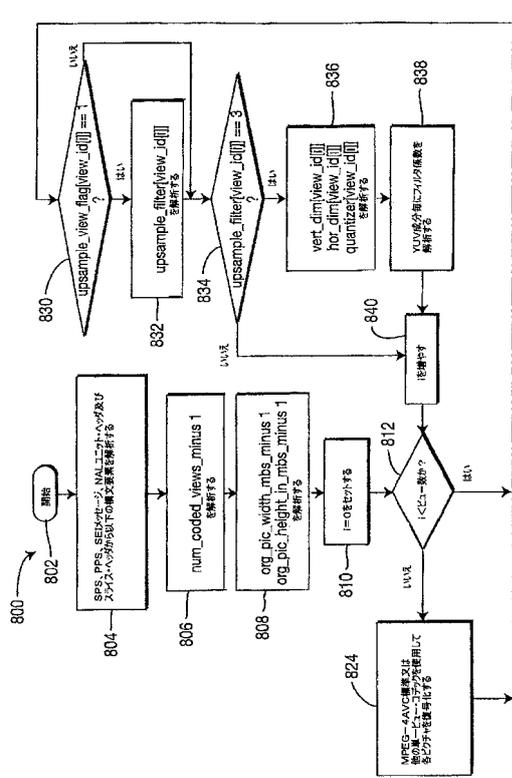
【図 7 A】



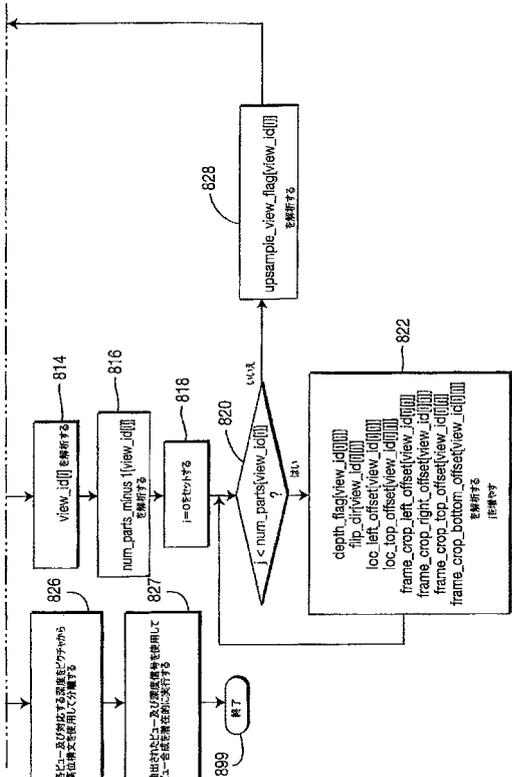
【図 7 B】



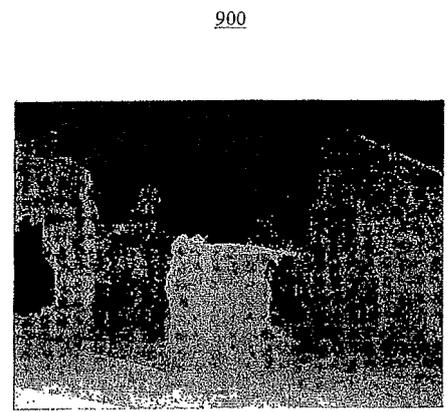
【図 8 A】



【図 8 B】

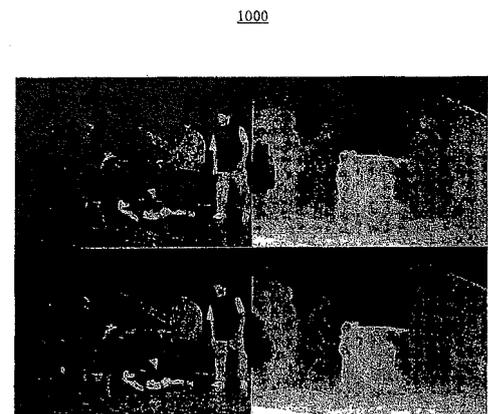


【図 9】



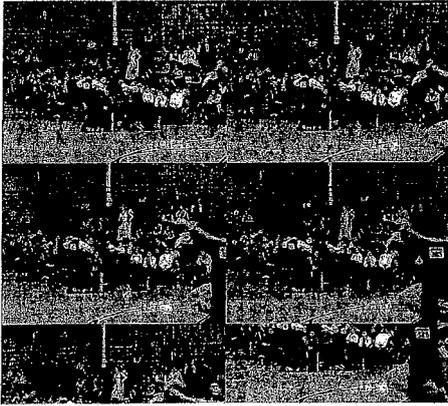
900

【図 10】



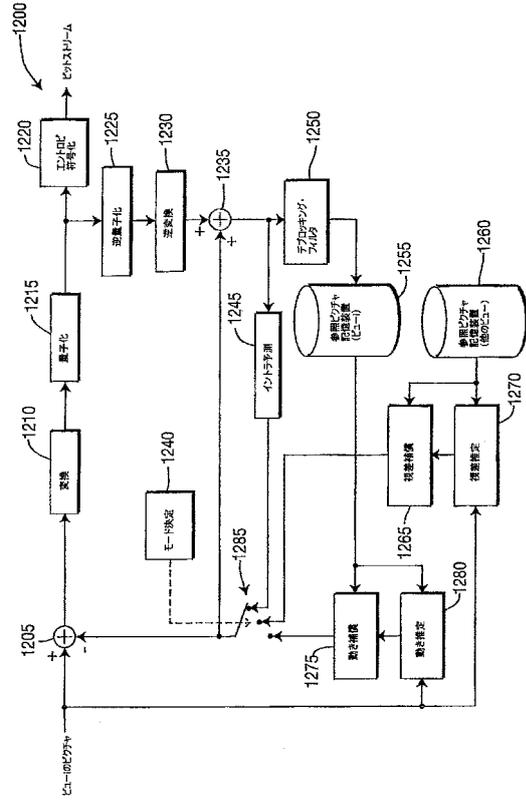
1000

【図 1 1】

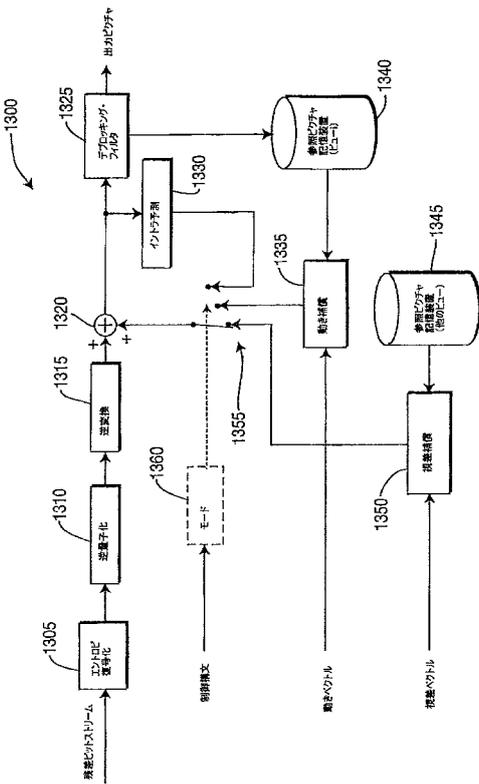


1100

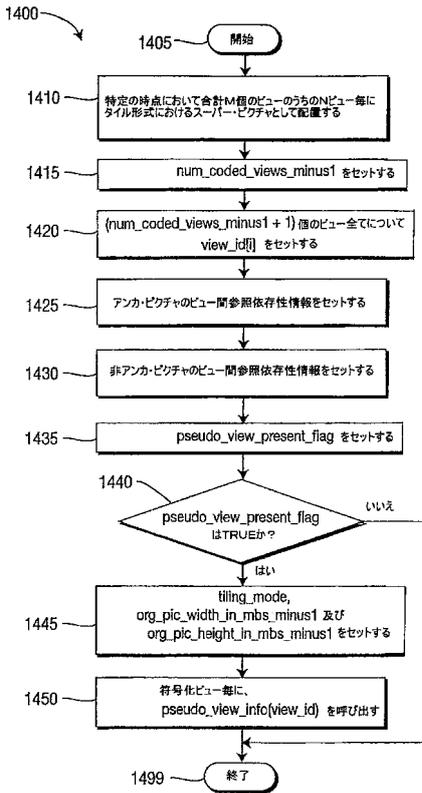
【図 1 2】



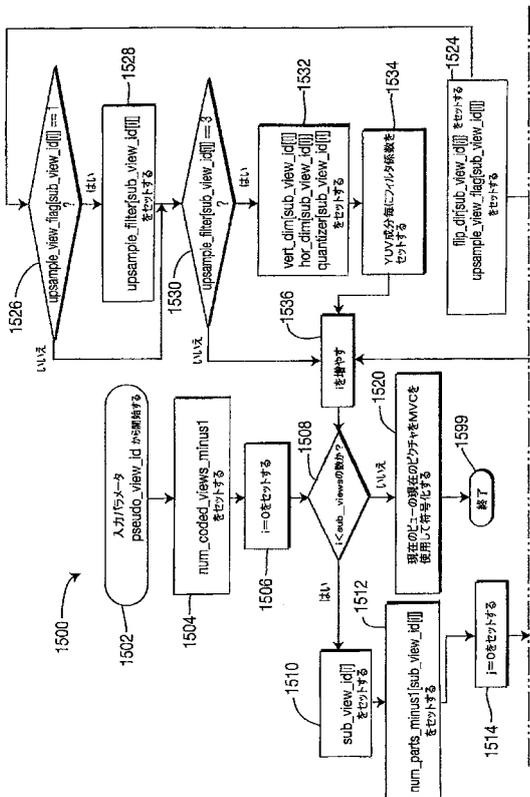
【図 1 3】



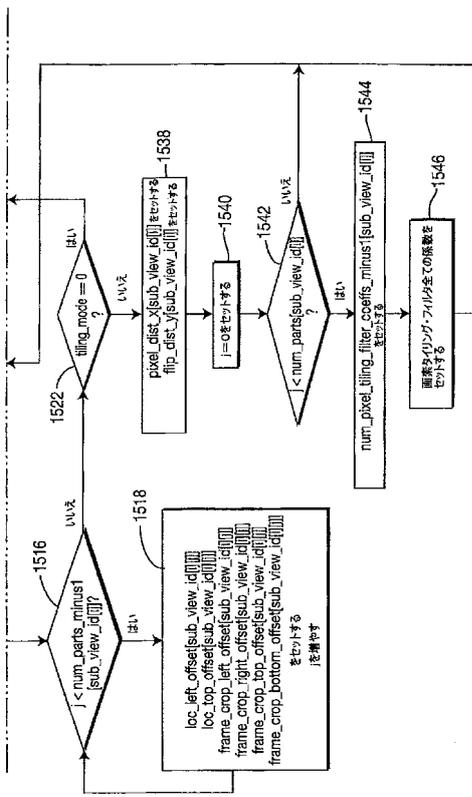
【図 1 4】



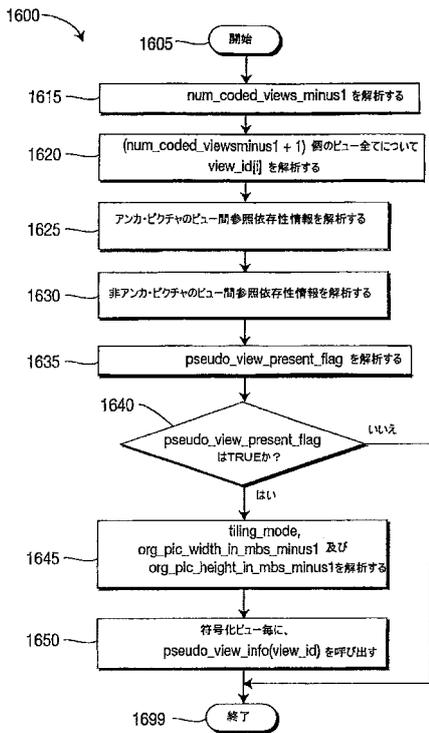
【図 15 A】



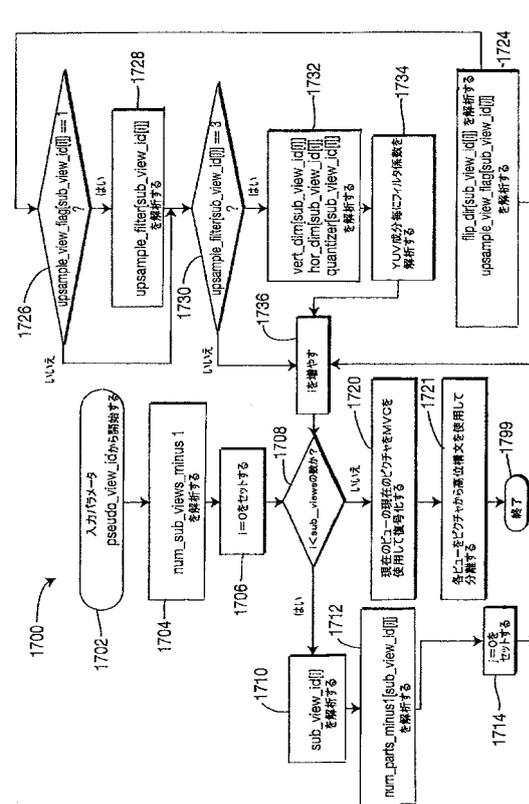
【図 15 B】



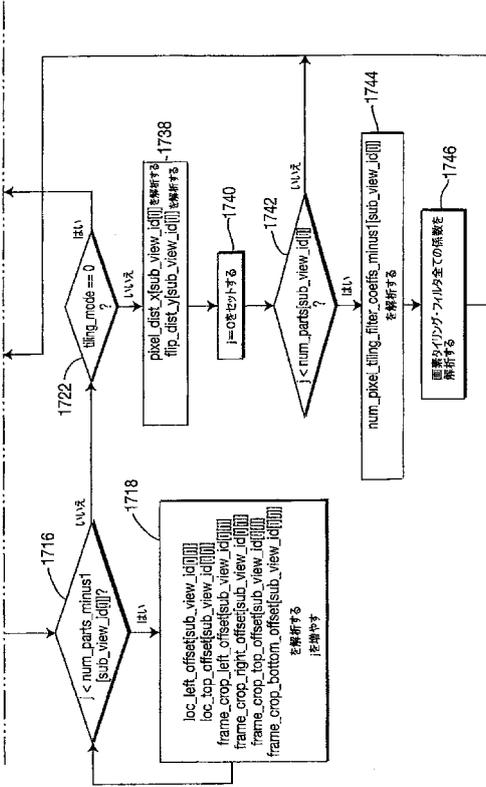
【図 16】



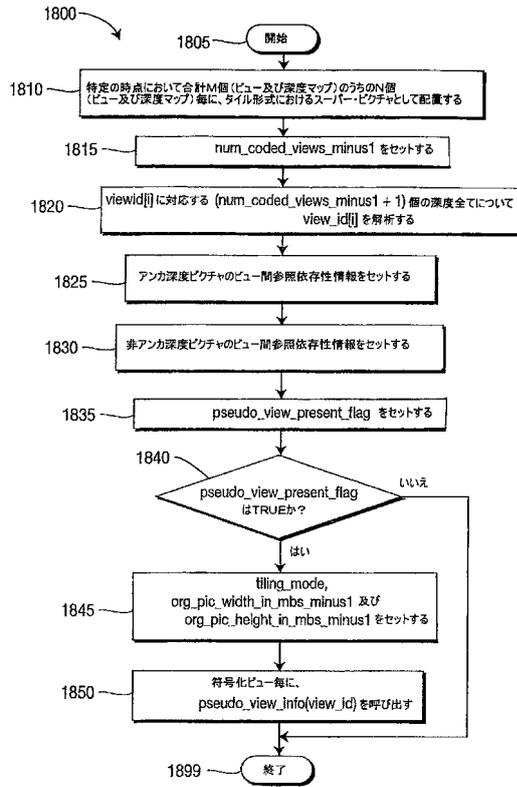
【図 17 A】



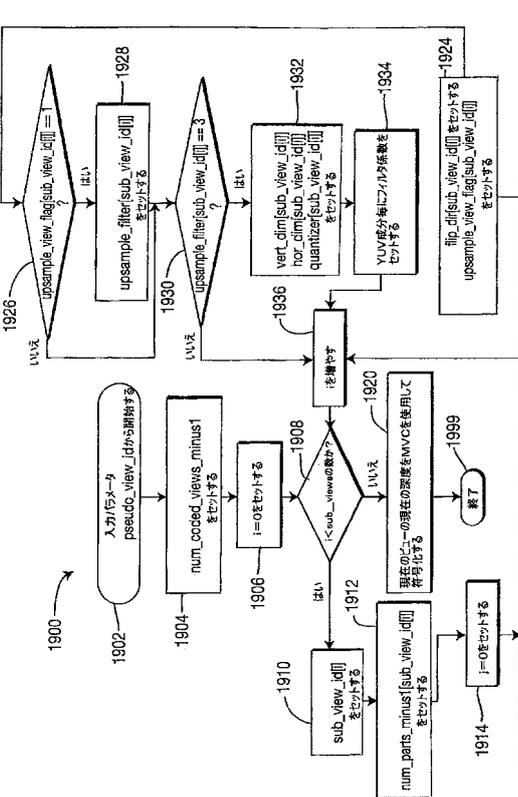
【図 17B】



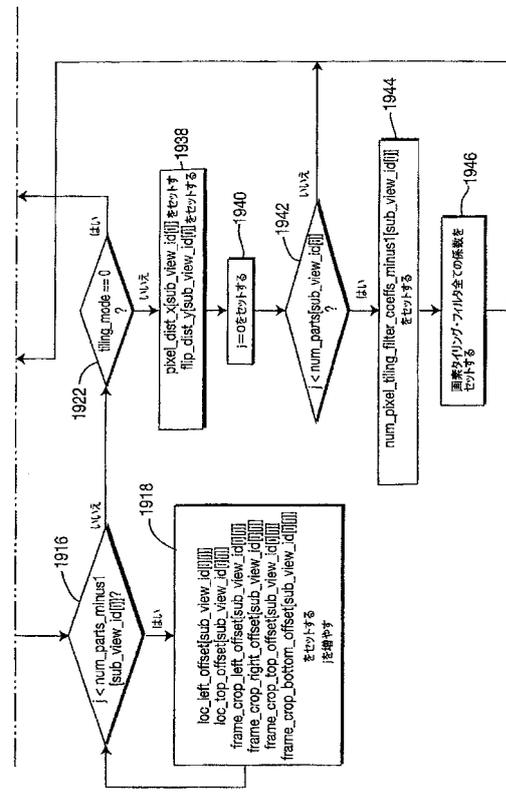
【図 18】



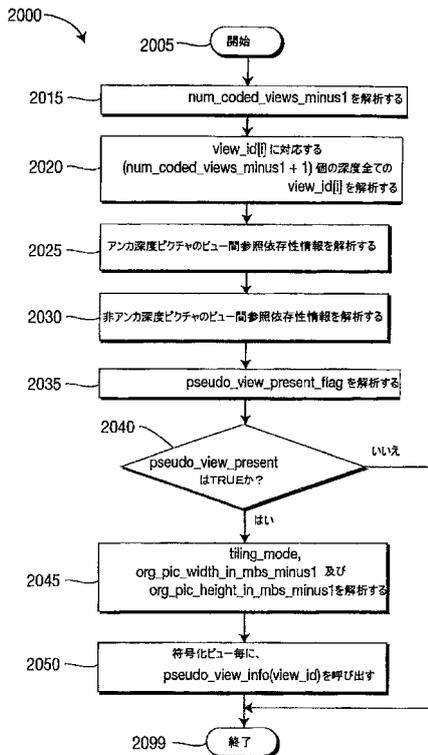
【図 19A】



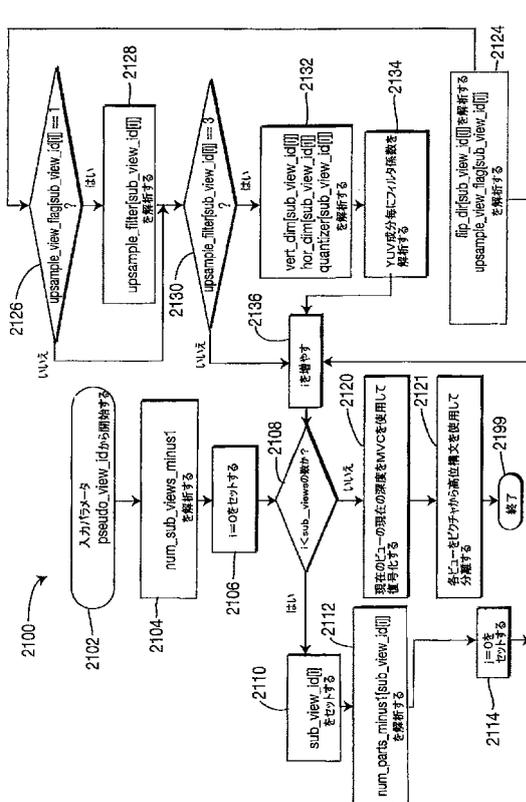
【図 19B】



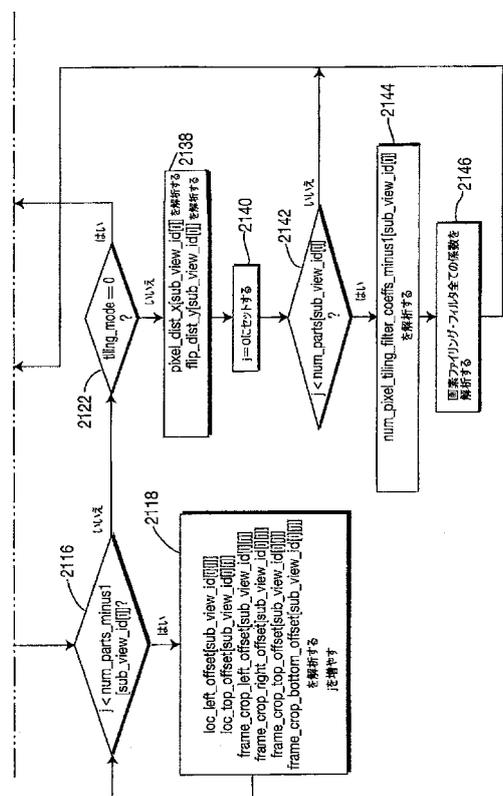
【図20】



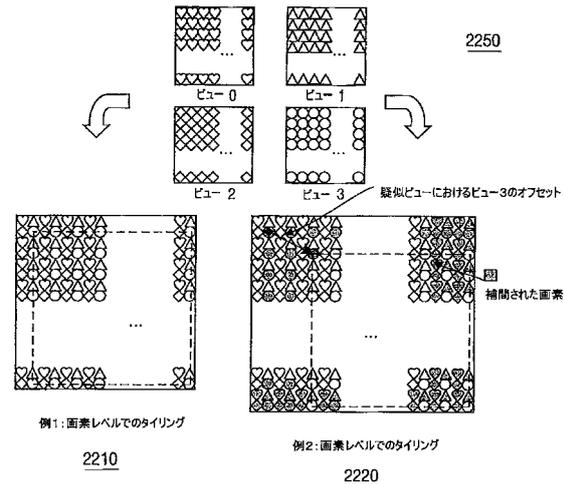
【図21A】



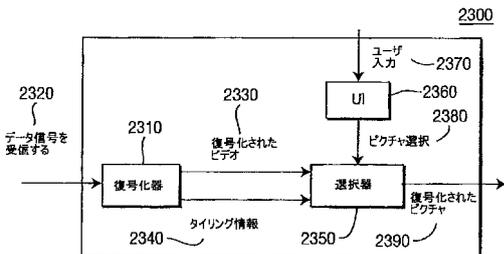
【図21B】



【図22】



【図23】



フロントページの続き

(72)発明者 パンディット, パーヴィン, ビバス
アメリカ合衆国, ニュージャージー州 0 8 8 2 3, フランクリン・パーク, ペア・トゥリー・レ
ーン 2 3

(72)発明者 イン, ペン
アメリカ合衆国, ニュージャージー州 0 8 5 5 0, ウエスト・ウィンザー, ウォーウィック・ド
ライヴ 6 5

(72)発明者 ティアン, ドン
アメリカ合衆国, ニュージャージー州 0 8 5 5 0, ウエスト・ウィンザー, ヘザー・ドライヴ
2 0 2 1 4

Fターム(参考) 5C159 KK37 KK38 LB05 LB15 MA16 PP03 PP13 RC11 UA02 UA05

【外国語明細書】

2019201435000001.pdf