

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第6637900号  
(P6637900)

(45) 発行日 令和2年1月29日(2020.1.29)

(24) 登録日 令和1年12月27日(2019.12.27)

(51) Int.Cl.	F 1
HO4N 19/895	(2014.01)
HO4N 19/174	(2014.01)
HO4N 19/436	(2014.01)
HO4N 19/44	(2014.01)
	HO4N 19/895
	HO4N 19/174
	HO4N 19/436
	HO4N 19/44

請求項の数 5 (全 16 頁)

(21) 出願番号	特願2016-559928 (P2016-559928)	(73) 特許権者	314015767 マイクロソフト テクノロジー ライセンシング、エルエルシー アメリカ合衆国 ワシントン州 98052 レッドモンド ワン マイクロソフト ウェイ
(86) (22) 出願日	平成27年3月26日(2015.3.26)	(74) 代理人	100107766 弁理士 伊東 忠重
(65) 公表番号	特表2017-516364 (P2017-516364A)	(74) 代理人	100070150 弁理士 伊東 忠彦
(43) 公表日	平成29年6月15日(2017.6.15)	(74) 代理人	100091214 弁理士 大貫 進介
(86) 國際出願番号	PCT/US2015/022599		
(87) 國際公開番号	W02015/153237		
(87) 國際公開日	平成27年10月8日(2015.10.8)		
審査請求日	平成30年2月22日(2018.2.22)		
(31) 優先権主張番号	14/231,228		
(32) 優先日	平成26年3月31日(2014.3.31)		
(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】 不完全なビデオデータのフレームの記録及び合成

## (57) 【特許請求の範囲】

## 【請求項 1】

計算システムにより実行されるコンピュータ実施の方法であって、前記計算システムはコンピュータ実行可能な命令を含むメモリを含み、前記命令は前記計算システムの一以上のプロセッサにより実行されると、不完全なビデオデータを復号する方法を実行し、前記コンピュータ実施の方法は、

ビデオデータに含まれるフレームの少なくとも1つのピースにアクセスすることであつて、前記フレームはそれが個別に復号可能な複数のフレームピースを含む、ことと、

前記フレームの少なくとも1つの追加フレームピースを受信されるのを待つ間に、アクセスされた少なくとも1つのフレームピースを復号することと、

前記フレームの少なくとも1つの追加フレームピースが受信されていないと判断することにより、前記フレームが不完全であると判断することと、

前記フレームが不完全であるとの判断及び指定された最小閾値数のフレームピースが受信されたとの判断に応じて、

少なくとも1つの前のフレームに関連するフレームデータの少なくとも一部を用いて、前記不完全なフレームの一以上のフレームピースを合成することと、

アクセスされ復号された少なくとも1つのフレームピースと、前記合成された一以上のフレームピースとを組み合わせることにより、前記不完全なフレームを完成させることと、

前記不完全なフレームが不完全であるとの判断及び前記指定された最小閾値数のフレー

10

20

ムピースが受信されていないとの判断に応じて、

フレーム置換データが送信されるべきことを示す通知を生成することと、

生成された通知を第2の計算システムに送信することと  
を含む、コンピュータ実施の方法。

**【請求項2】**

インクリメント状に増加するタイムスタンプを用いて、フレームとフレームピースが別々にトラッキングされる、請求項1に記載のコンピュータ実施の方法。

**【請求項3】**

前記不完全なフレームの前方エラー訂正データを待つことと、前記前方エラー訂正データを待っている間に、前記不完全なフレームの他のフレームピースを復号し続けることと  
を含む、  
10 請求項1に記載のコンピュータ実施の方法。

**【請求項4】**

前記不完全なフレームの前方エラー訂正データを受け取ることと、  
前記前方エラー訂正データが対応する前記不完全なフレームのフレームピースを復号す  
ることとをさらに含む、  
15 請求項3に記載のコンピュータ実施の方法。

**【請求項5】**

計算システムに請求項1乃至4いずれか一項に記載のコンピュータ実施の方法を実行さ  
せるコンピュータプログラム。  
20

**【発明の詳細な説明】**

**【背景技術】**

**【0001】**

世界中で日常的に、インターネットその他のネットワークによりコンシューマに、ビデ  
オデータが送信されている。ビデオデータは一般的にビデオフレームデータパケットロス  
を許容する、又は補償するプロトコルを用いて送信される。例えば、H.264フォーマ  
ットを用いて符号化されたビデオフレームは、異なる複数のピースすなわち「スライス」  
を用いて送信されてもよく、各スライスは複数のフレームデータパケットを含んでもよい  
。これらのパケットは、伝送中に失われたり、様々な理由で遅延したりすることがある。  
あるフレームのフレームデータがある時間にわたり受信されないままである場合、そのフ  
レームは破棄され、視覚的忠実性が損なわれ、ビデオコンシューマにより知覚され  
ることがある。  
30

**【発明の概要】**

**【0002】**

ここに説明する実施形態は、不完全なビデオデータの復号に関する。一実施形態では、  
コンピュータシステムはフレームのピース(a piece of a frame)(  
例えば、ビデオフレーム)を受信する。フレームは、複数のフレームピース又は「スライ  
ス」を含み、各フレームピースは個別に復号可能であるように符号化される。コンピュー  
タシステムは、少なくとも1つの追加フレームピースを受信するのを待ちつつ、受信した  
フレームピースを復号する。次いで、フレームが不完全であることを示す状態が発生する  
と、コンピュータシステムはフレームピースを合成して、まだ受信されていない追加フレ  
ームピースを置き換える。  
40

**【0003】**

他の一実施形態において、フレームのピースの受信に応じて、コンピュータシステムは  
、他のフレームピースが受信されるのを待ちつつ、受信したフレームピースを復号する。  
コンピュータシステムは、フレームが不完全であり、そのフレームには少なくとも1つの  
まだ受信される予定の追加フレームピースがあることを示す状態の発生を検出する。次い  
で、その発生の検出に応じて、コンピュータシステムは、まだ受信される予定の追加フレ  
ームを置き換えるフレームピースを合成する。

**【0004】**

10

20

30

40

50

他の一実施形態において、コンピュータシステムは、複数のフレームピースを含むフレームの1つのフレームピースにアクセスする。各フレームピースは個別に復号可能である。コンピュータシステムは、追加フレームピースが受信されるのを待ちつつ、アクセスしたフレームピースを復号する。コンピュータシステムは、少なくとも1つの追加フレームピースがまだ受信される予定であると判断し、その判断に応じて、前のフレームに関連するフレームデータを用いてフレームピースを合成する。

#### 【0005】

本欄では、発明の詳細な説明で詳しく説明するコンセプトの一部を選んで、簡単に説明する。本欄は、特許を請求する主題の重要な特徴や本質的な特徴を特定するものではなく、特許を請求する主題の範囲を決定するのを助けるものでもない。

10

#### 【0006】

その他のフィーチャと利点は、以下の説明に記載され、部分的に本技術分野の当業者には説明により明らかであり、又はここでの教示の実施により学習されてもよい。ここに説明の実施形態のフィーチャと利点は、添付した特許請求の範囲に指摘される機器及び組み合わせにより実現され取得されてもよい。ここに説明の実施形態のフィーチャは、以下の説明及び添付した特許請求の範囲からより完全に明らかになるだろう。

#### 【図面の簡単な説明】

#### 【0007】

ここに説明する上記その他のフィーチャ(features)をさらに明確にするため、添付の図面を参照して、より具体的な説明をする。言うまでもなく、これらの図面はここに説明する実施形態の例を示しているだけであり、実施形態の範囲を限定するものであると考えてはならない。添付した図面を用いて、実施形態をさらに具体的に詳しく説明する。

20

【図1】ここに説明する実施形態が、不完全なビデオデータの復号も含めて、動作してもよいコンピュータアーキテクチャを示す図である。

【図2】不完全なビデオデータを復号する方法の一例を示すフローチャートである。

【図3】不完全なビデオデータを復号する方法の別の一例を示すフローチャートである。

【図4】不完全なビデオデータを復号する方法のもう一つ別の一例を示すフローチャートである。

【図5】フレームピースが順次的に復号される一実施形態を示す図である。

30

【図6】フレームピースが復号され合成される一実施形態を示す図である。

【図7】フレームピースがH.264チャネルで復号される一実施形態を示す図である。

#### 【発明を実施するための形態】

#### 【0008】

ここに説明する実施形態は、不完全なビデオデータの復号に関する。一実施形態では、コンピュータシステムはフレームのピース(a piece of a frame)(例えば、ビデオフレーム)を受信する。フレームは、複数のフレームピースを含み、各フレームピースは個別に復号可能であるように符号化される。コンピュータシステムは、少なくとも1つの追加フレームピースを受信するのを待ちつつ、受信したフレームピースを復号する。次いで、フレームが不完全であることを示す状態が発生すると、コンピュータシステムは、まだ受信されていない追加フレームピースを置き換えるフレームピースを合成する。

40

#### 【0009】

他の一実施形態では、フレームのピースの受信に応じて、コンピュータシステムは、他のフレームピースが受信されるのを待ちつつ、受信したフレームピースを復号する。コンピュータシステムは、フレームが不完全であり、そのフレームには少なくとも1つのまだ受信される予定の追加フレームピースがあることを示す状態の発生を検出する。次いで、その発生の検出に応じて、コンピュータシステムは、まだ受信される予定の追加フレームを置き換えるフレームピースを合成する。

#### 【0010】

50

他の一実施形態では、コンピュータシステムは、複数のフレームピースを含むフレームの1つのフレームピースにアクセスする。各フレームピースは個別に復号可能である。コンピュータシステムは、追加フレームピースが受信されるのを待ちつつ、アクセスしたフレームピースを復号する。コンピュータシステムは、少なくとも1つの追加フレームピースがまだ受信される予定であると判断し、その判断に応じて、前のフレームに関連するフレームデータを用いてフレームピースを合成する。

#### 【0011】

以下の説明では、実行してもよい複数の方法及び方法動作に言及する。留意点として、方法動作はある順序で説明され、又はある順序で行うとフローチャートで例示するが、特に断らなければ、特定の順序は必ずしも必要ない。または、ある動作は、それが実行される前に他の動作が完了していることに依存するため、特定の順序が必要となる。10

#### 【0012】

ここに説明する実施形態は、様々なタイプの計算システムを実施してもよい。これらの計算システムは、ますますバラエティに富んだ形式を取っている。計算システムは、例えば、ハンドヘルドデバイス、電化製品、ラップトップコンピュータ、デスクトップコンピュータ、メインフレーム、分散コンピューティングシステム、又は従来は計算システムと考えられていなかったデバイスであってもよい。この説明及びクレームにおいて、「計算システム」との用語は、少なくとも1つの物理的かつ有体のプロセッサと、プロセッサにより実行されてもよいコンピュータ実行可能命令を格納することができる物理的かつ有体のメモリとを含む任意のデバイス又はシステム（又はこれらの組み合わせ）を含むものと広く定義される。計算システムは、ネットワーク環境にわたり分散していくてもよく、構成要素である複数の計算システムを含んでいてもよい。20

#### 【0013】

図1に示したように、計算システム101は、一般的に、少なくとも1つの処理部102Aとメモリ103Aとを含む。メモリ103Aは物理的システムメモリであってもよく、これは揮発性、不揮発性、又は両者の組み合わせであってもよい。「メモリ」との用語は、ここでは、物理的記憶媒体などの不揮発性マストレージを指して使われても良い。計算システムが分散している場合、処理、メモリ、及び／又は記憶機能も分散していくてもよい。

#### 【0014】

ここで、「実行可能モジュール」又は「実行可能コンポーネント」との用語は、ソフトウェアオブジェクト、ルーチン、又は方法であって計算システム上で実行してもよいものを指し得る。ここに説明する異なるコンポーネント、モジュール、エンジン、及びサービスは、（例えば、別のスレッドとして）計算システム上で実行されるオブジェクト又はプロセスとして実施してもよい。30

#### 【0015】

以下の説明では、実施形態を、一以上の計算システムにより実行される動作を参照して説明する。かかる動作がソフトウェアで実装されている場合、その動作を実行する計算システムの一以上のプロセッサは、コンピュータ実行可能命令の実行に応じて、計算システムの動作を命じる。例えば、かかるコンピュータ実行可能命令は、コンピュータプログラム製品を構成する一以上のコンピュータ読み取り可能媒体に実施してもよい。かかる動作の一例はデータの操作を含む。コンピュータ実行可能命令（及び操作されたデータ）は、計算システム101のメモリ103Aに格納されてもよい。計算システム101は、それが有線又は無線ネットワークにより他のメッセージプロセッサと通信できるようにする通信チャンネルを有していてもよい。40

#### 【0016】

ここに説明する実施形態は、以下により詳しく説明するように、例えば一以上のプロセッサとシステムメモリなどのコンピュータハードウェアを含む特定目的又は汎用コンピュータシステムを含み、又は利用してもよい。システムメモリは全体メモリ103A内に含まれていてもよい。システムメモリは、「メインメモリ」とも呼ばれ、メモリバスにより50

少なくとも 1 つの処理部 102A によりアドレッシング可能なメモリロケーションを含む。この場合、アドレスロケーションはメモリバス自体によってアサートされる。システムメモリは、従来は揮発性であったが、ここに説明する原理は、システムメモリが部分的に又は完全に不揮発性である場合にも当てはまる。

#### 【 0 0 1 7 】

また、本発明の範囲内の実施形態は、コンピュータ実行可能命令及び／又はデータ構造を担う又は格納する物理的その他のコンピュータ読み取り可能媒体を含む。かかるコンピュータ読み取り可能媒体は、汎用又は特殊目的コンピュータシステムによりアクセスできる、利用できるどんな媒体であってもよい。コンピュータ実行可能命令及び／又はデータ構造を格納するコンピュータ読み取り可能媒体は、コンピュータ記憶媒体である。コンピュータ実行可能命令及び／又はデータ構造を担うコンピュータ読み取り可能媒体は、伝送媒体である。このように、限定ではなく例として、本発明の実施形態は、区別でき異なる少なくとも 2 種類のコンピュータ読み取り可能媒体、すなわち、コンピュータ記憶媒体と伝送媒体とを含み得る。10

#### 【 0 0 1 8 】

コンピュータ記憶媒体は、コンピュータ実行可能命令及び／又はデータ構造を記憶する物理的ハードウェア記憶媒体である。物理的ハードウェア記憶媒体は、RAM、ROM、EEPROM、半導体ドライブ(SSD)、フラッシュメモリ、相変化メモリ(PCM)、光ディスクストレージ、磁気ディスクストレージその他の磁気ストレージデバイス、又はその他の任意のハードウェアストレージデバイスであってコンピュータ実行可能命令又はデータ構造の形式のプログラムデータを記憶するのに使え、汎用又は特殊用途コンピュータシステムによりアクセス及び実行でき、本発明の開示の機能を実施できるものなどのコンピュータハードウェアを含む。20

#### 【 0 0 1 9 】

送信媒体は、コンピュータ実行可能命令又はデータ構造の形式のプログラムコードを担うのに用いることができ、汎用又は特殊用途コンピュータシステムによりアクセスできるネットワーク及び／又はデータリンクを含み得る。「ネットワーク」はコンピュータシステム及び／又はモジュール及び／又はその他の電子デバイス間での電子データの送信を可能にする一以上のデータリンクとして定義される。ネットワークその他の通信接続(有線、無線、又は有線と無線の組み合わせを問わない)によりコンピュータシステムに情報が転送又は供給されるとき、コンピュータシステムは、その接続を伝送媒体として見ることがある。上記のものの任意の組合せも、コンピュータ読み取り可能媒体の範囲内に含まれる。30

#### 【 0 0 2 0 】

さらに、様々なコンピュータシステムコンポーネントに到達するとき、コンピュータ実行可能命令又はデータ構造の形式のプログラムコードは、伝送媒体からコンピュータストレージ媒体に(又はその逆に)自動的に転送され得る。例えば、ネットワーク又はデータリンクにより受信したコンピュータ実行可能命令又はデータ構造は、ネットワークインターフェースモジュール(例えば、NIC)中のRAMにバッファでき、その後最終的にコンピュータシステムのコンピュータシステムRAM及び／又は揮発性の低いコンピュータストレージ媒体に転送される。このように、言うまでもなく、コンピュータストレージ媒体は、伝送媒体も(又は伝送媒体を主に)利用するコンピュータシステムコンポーネントに含まれ得る。40

#### 【 0 0 2 1 】

コンピュータ実行可能命令は、例えば、命令とデータであって、一以上のプロセッサで実行されると、汎用コンピュータシステム、特殊用途コンピュータシステム、又は特殊用途処理デバイスに、ある機能又は一群の機能を実行させるものを含む。コンピュータ実行可能命令は、例えば、バイナリ、アセンブリ言語などの中間フォーマット命令、又はソースコードなどであってもよい。

#### 【 0 0 2 2 】

10

20

30

40

50

本技術分野の当業者には言うまでもなく、ここに説明の原理は、多くのタイプのコンピュータシステム構成を有するネットワークコンピューティング環境で実施してもよい。コンピュータシステム構成には、パーソナルコンピュータ、デスクトップコンピュータ、ラップトップコンピュータ、メッセージプロセッサ、ハンドヘルドデバイス、マルチプロセッサシステム、マイクロプロセッサベース又はプログラマブルコンシューマエレクトロニクス、ネットワークPC、ミニコンピュータ、メインフレームコンピュータ、携帯電話、PDA、タブレット、ページャ、ルータ、スイッチなどを含む。本発明は、ローカル及びリモートコンピュータシステムがネットワークを通してリンク（有線データリンク、無線データリンク、又は有線データリンクと無線データリンクの組み合わせ）されていて、それらが両方ともタスクを実行する分散システム環境で実施してもよい。そのため、分散システム環境では、コンピュータシステムは構成要素の複数のコンピュータシステムを含んでいてもよい。分散システム環境では、プログラムモジュールは、ローカル及びリモートのメモリ記憶デバイスの両方に配置できる。10

#### 【0023】

本技術分野の当業者には言うまでもなく、本発明はクラウドコンピューティング環境で実施してもよい。クラウドコンピューティング環境は分散していてもよいが、これは必ずしも必要ない。分散している場合、クラウドコンピューティング環境は、ある組織内で国際的に分散していてもよく、複数の組織にまたがって所有されるコンポーネントを有していても良い。この説明及び以下の請求項において、「クラウドコンピューティング」は、構成可能な計算リソース（例えば、ネットワーク、サーバ、ストレージ、アプリケーション、及びサービス）の共有プールへのオンデマンドネットワークアクセスを可能にするモデルとして定義される。「クラウドコンピューティング」の定義は、適切に展開されたとき、かかるモデルから得られる他の多くの利点のどれにも限定されない。20

#### 【0024】

さらにまた、ここに説明のシステムアーキテクチャは、それぞれがシステムの全体としての機能に貢献する、独立した複数のコンポーネントを含み得る。このモジュール性により、プラットフォームスケーラビリティの問題にアプローチするときの柔軟性が改善し、この目的を達成するため、様々な利点を提供する。機能範囲が限定されたより小さいスケールのパートを用いることにより、システムの複雑性及び成長をより容易に管理することができる。これらの弱く結合したモジュールの使用により、プラットフォームの耐故障性が向上する。個々のコンポーネントは、ビジネスニーズに応じて、徐々に成長できる。モジュール開発は、新しい機能のタイムツーマーケットの短縮にもつながる。コアシステムに影響を与えずに、新しい機能を追加又は削除できる。30

#### 【0025】

図1は、少なくとも1つの実施形態が利用され得るコンピュータアーキテクチャ100を示す。コンピュータアーキテクチャ100は、コンピュータシステム101を含む。コンピュータシステム101と113は、任意のタイプのローカル又は分散コンピュータシステムであってもよく、クラウドコンピューティングシステムを含んでいてもよい。それ少なくとも1つのプロセッサ102A/102B、メモリ103A/103B及び通信モジュール104A/104Bを含む。通信モジュールは、有線又は無線ネットワークカード、ブルートゥース(登録商標)無線通信部、Wi-Fi無線通信部、又はその他のデジタルデータを送受信するように構成されたハードウェアを含む有線又は無線通信手段を含んでいてもよい。通信モジュール104Aは、例え、コンピュータシステム113の通信モジュール104Bから、フレーム111又はフレームピース110を受信してもよい。40

#### 【0026】

幾つかの実施形態では、コンピュータシステム113はサーバであってもよい。サーバは、単一のコンピュータシステムであってもよく、分散していてもよい。サーバは、データを、コンピュータシステム101などのクライアントに提供するように構成されていてもよい。サーバは、データの入力又はその他の要求に応じて、クライアントにアプリケー50

ションデータを提供してもよい。幾つかの実施形態では、コンピュータシステム 113 はビデオゲームサーバであってもよい。かかる場合、ビデオゲームサーバは、ビデオゲームコンテンツを有するフレームを提供するように構成されていてもよい。これらのフレームは、符号化されていてもよく、少なくとも幾つかの場合には、原データ (raw form) であってもよい。フレームは、フルフレーム形式 111 又はピース 110 で送信されてもよい。ピース (pieces) はデータパケットを介して送信されてもよく、各ピースは一以上のデータパケットを含んでいてもよい。そのため、フレームピースはコンピュータシステム 101 に転送され、次いで結合されてフルフレームを形成する。これらのフレームピースは、動画を構成する連続した一連のフレームを構成するため、連続したストリーミング方式で送信されてもよい。ゲームシナリオでは、このフレームストリームはビデオゲーム出力を構成するだろう。言うまでもなく、ビデオゲームの実施形態は、ここでは頻繁に言及するが、ビデオコンテンツが 1 つのコンピュータシステムから他のコンピュータシステムに転送される（すなわち、ストリーミングされる）単なる一実施形態の一例である。実際、不完全なデータの復号は、以下に説明するように、ビデオデータがロッシャー (lossy) なネットワークにより送信される異なる様々なシナリオで実装してもよい。

#### 【0027】

フレーム 111 とフレームピース 110 を転送しやすくするために、これらは転送前に何らかの方法で符号化されることが多い。幾つかの実施形態では、フレームは H.264 フォーマットにより符号化されてもよい。他の実施形態では、他のフォーマットを用いても良い。コンピュータシステム 113 の符号化モジュール 114 は、選択された符号化モードマップにより、フルフレーム及び / 又はフレームピース 110 を符号化してもよい。これらの符号化されたフレーム又はフレームピースは、コンピュータシステム 101 に転送され、そこで復号され、ディスプレイ 108 に表示される。復号モジュール 105 は、受信したフレーム 111 及びフレームピース 110 を復号し、ユーザに表示するためディスプレイ 108 に送る。幾つかの実施形態では、復号モジュール 105 は、幾つかのフレームピースが失われても、フレームピース 110 を表示するように構成されていてもよい。実際、データパケットが失われるネットワークトランザクションでは、幾つかのフレームピース又はその一部が失われ、又は転送中に遅延することがある。かかる場合に、復号モジュールは、失われた又は遅延したピースが現れるのを待ちつつ、受信したフレームピース 110 を復号し続ける。失われた又は遅延したピースが現れるのが間に合わないとき、フレーム合成モジュール 107 は、様々なフレーム合成手法を用いてフレームを合成してもよい。

#### 【0028】

例えば、フレーム合成モジュール 107 は、その前の 1 以上のフレームのフレームデータを見て、失われた又は遅延したフレームピースを合成してもよい。フレーム合成モジュール 107 は、失われたフレームピースが、現在復号されており、表示する準備がされているフレームの左上角にあると判断してもよい。フレーム合成モジュール 107 は、前の 1 以上のフレームの左上角のフレームデータを見て、そのフレームのその部分がどう見えるか（例えば、青空か、都市背景など）確認する。次いで、このデータに基づいて、フレーム合成モジュール 107 は、失われたフレームピースを合成して、フレームがほぼ正しく見えるようにしてもよい。しかし、忠実性は幾分失われる可能性がある。この点において、コンピュータシステム 101 は、コンピュータシステム 113 に、少なくとも 1 つのフレームピースが失われた又は大幅に遅延したこと、及びフルフレーム 111 が送信されることを示す表示 (indication) 112 を送信してもよい。次いで、フレーム差分を送信（すなわち、変化したフレームの部分のみを送信）する替わりに、フルフレーム 111 が送信され、又は正しくないピクセルを修正する他の何らかのリフレッシュ (refresh) が行われる。留意点として、他の多くの方法や手法を用いて失われたフレームピースを合成してもよい。これらのフレーム合成方法その他のコンセプトは、図 2、3 及び 4 のそれぞれ方法 200、300 及び 400 を参照して、以下により詳しく説明

10

20

30

40

50

する。

#### 【0029】

以下に説明するシステム及びアーキテクチャを考慮して、開示の手段により実施されてもよい方法は、図2、3及び4のフローチャートを参照してより良く理解されるだろう。説明を単純にするために、方法を一連のブロックとして図示して説明する。しかし、言うまでもなく、請求項に係る主題は、ブロックの順序には限定されない。幾つかのブロックは、ここに図示して説明するものとは異なる順序で、及び／又は他のブロックと同時に行われてもよいからである。さらに、以下に説明する方法を実装するのに、例示されるブロックのすべてが必要ではないかも知れない。

#### 【0030】

図2は、不完全なビデオデータを復号する方法200を示すフローチャートである。環境100のコンポーネント及びデータを頻繁に参照して、方法200をここで説明する。

#### 【0031】

方法200は、少なくとも1ピースのフレームを受信する動作を含む。フレームは複数のフレームピースを含み、各フレームピース自体は、個別の復号可能なように、符号化される（動作210）。例えば、コンピュータシステム101の通信モジュール104Aは、コンピュータシステム113の通信モジュール104Bからフレームピース110を受信してもよい。フレームピース110はそれだけで送信されても、他のピースと組み合わせて送信されてもよい。さらに、フレームピース自体は单一のデータパケットで送信されてもよく、複数のデータパケットに分割されてもよい。フレームピース110は、フルフレーム（例えば、111）の1ピースであり、そのため、対応する他のフレームピースと組み合わせられると、フルフレームとなる。フレームは実質的に幾つのフレームピースに分割されてもよい。これらのピースは、転送され、再アセンブルされてフルフレームを形成する。幾つかの実施形態では、フレームピースは、コンピュータシステム101に転送される前に、符号化モジュール114により符号化されてもよい。フレームは異なるさまざまな符号化フォーマットのどれで符号化されてもよい。これらの符号化フォーマットは、画質、転送効率（すなわち、帯域幅）、符号化の速さ、又はその他の点を考慮して最適化されてもよい。

#### 【0032】

各フレームピース110は、そのフレームピースが個別に復号できるように、個別に符号化されてもよい。このように、コンピュータシステム101の復号モジュールは、フレームピースを受け取り、それを他のフレームピースを受信せずに復号してもよい。次いで、他のフレームピースを受信後、他のフレームピースの受信を待ちつつ、復号モジュール105は、最も新しく受信したピースの復号を開始してもよい。実際、これは方法200で説明した。方法200は、少なくとも1つの追加のフレームピースが受信されるのを待ちつつ、受信したフレームピースの少なくとも1つを復号する動作を含む。このプロセスはさらに図5に示した。

#### 【0033】

図5は、図1の復号モジュール105と同じでもよく、異なってもよいデコーダ501を含む。デコーダ501は、フレームピース503Aを前に受信し、その復号をすでに開始していてもよい。この復号されるフレームピース503Aは、追加されてもよいし、フレーム505の構成を開始してもよい。フレーム505の下部の構成として示したが、言うまでもなく、復号されるフレームピース503は、フレームの実質的にどの部分のフレームデータを含んでもよく、フレーム中の1つのブロックのフレームデータを含んでもよく、フレームの異なる部分のフレームデータを含んでいてもよい。しかし、単純化のため、図5と図6に示したフレームピースは、互いに水平に重なっている。図5に示した、入来するフレームピース502B、502C及び502Dは、トランジットであり、デコーダ501によりまだ受信されていない。しかし、これらのフレームピースが受信されると、デコーダ501はそれらの復号を個別に開始できる。このように、受信されたフレームピース503Aは、フレームピース502Bが受信されるのを待っている間に、復号され

10

20

30

40

50

得る。同様に、フレームピース 503B は、フレームピース 502C が受信されるのを待っている間に、復号でき、フレームピース 503C は、フレームピース 502D が受信されるのを待っている間に復号でき、以下同様である。このように、デコーダ 501 は、現在有しているフレームデータを用いて復号プロセスを開始することができる。

#### 【0034】

フレームピースは順序通りでなく復号することもできる。フレームピースは、一般的にロッシー (lossy) ネットワークにより転送され、各パケットの安全な伝送を補償しないプロトコル（例えば、UDP）を用いて転送されてもよい。そのため、フレームピースは、様々な理由で失われたり、遅延したりすることがある。一例では、フレームピース 502C がフレームピース 502B より前に受信されるかも知れない。このような場合、デコーダ 501 は、フレームピース 502B を受信する前に、フレームピース 502C の復号を開始してもよい。次いで、フレームピース 502B が受信されると、復号され、フレーム 505 に追加される。最後に、フレーム 505 のすべてのコンポーネントピースが受信されると、フレームがディスプレイ 504 に表示される。ディスプレイは任意のタイプのモニタ、テレビジョン、タブレット、又は表示手段であってもよい。フレーム 505 は、一連のフレーム中の 1 つのフレームであってもよい。例えば、ストリーミングビデオ又はビデオゲームのシナリオでは、データストリーミングサーバが、かかるフレームを毎秒多数送信する。これらのフレームは、ディスプレイ 504 上に順次的に表示され、ユーザが見る動画を構成する。

#### 【0035】

図 2 に戻り、方法 200 は、フレームが不完全であることを示す少なくとも 1 つの状態が成立した場合、一以上のフレームピースを合成して、まだ受信されていない少なくとも 1 つの追加フレームを置き換える動作（動作 230）を含む。その状態は、例えば、一以上のフレームピースが失われ、又は遅延したとの判断を含んでいてもよい。状態モニタリングモジュール 106 は、入来するフレームピースをモニタするように構成され、フレームピースが失われたこと、又は大幅に遅延していることを判断してもよい。かかる判断ができると、フレーム合成モジュール 107 は、失われたフレームピースを合成して、合成したデータをフレームに挿入して、そのフレームを完成させてよい。

#### 【0036】

図 6 に示すように、復号されたフレームピースは、合成されたフレームピースと結合され、フレームを形成してもよい。デコーダ 601 は、図 5 を参照して上で説明したように、フレームピースを受信する。図示したように、フレーム 607 はすでに復号されたフレームピース 605A を含む。フレームピース 605A は、フレームピース 602B、602C、602D、602E 以下同様を待っている間に、復号されてもよい。図 1 の状態モニタリングモジュール 106 は、フレームピース 604B が失われ又は遅延したと判断し、フレーム合成モジュール 107 にフレームピース 604B の合成を開始させていてよい。合成されるデータは、図 3 を参照して下で説明するように、様々な方法で取得されてもよい。合成されたフレームピース 604B は、実際のフレームピースが挿入されるはずだったフレーム 607 の部分に、挿入されてもよい。デコーダ 601 は、合成器が他のフレームピースを合成している間に、フレームピースの復号を続けても良い。このように、フレームピース 602C は、受信され、復号され、復号フレームピース 605C となる。602D や 602E などの他のフレームピースも、合成器が失われた又は遅延したフレームピースのデータを合成している間に、受信され、復号される。

#### 【0037】

図 1 に戻って、状態モニタリングモジュール 106 は、フレーム 111 が不完全であることを示す状態が発生したことを判断してもよい。この状態は、指定された最小閾値数のフレームピースが受信され、フレームピース合成が行われることの表示を含んでもよい。このように、例えば、フレーム毎又は一連のフレーム毎に、フレームピースの最小数を決めてよい。フレームピースの最小数に達したら、そのフレームは合成により補正できると見なされてもよい。そのため、フレームは、残りのフレームピースの再生成（すなわち

10

20

30

40

50

、合成)が許される十分な受信データを含む。このフレームピースの最小数は、アドミニストレータ、プログラマ、又はその他のユーザにより設定されてもよい。

#### 【0038】

最小数のフレームピースが受信されていない場合、又は合成が可能ではない又は(例えば、多くのアーティファクトが発生してしまい)望ましくない場合、コンピュータシステム101は、フレーム置換データを送信するべきことを示す表示112を生成してもよい。少なくとも幾つかの実施形態では、フレーム置換データはフルフレーム(例えば、111)である。表示112が計算システム113に送信され、計算システム113はフルフレーム111を送信して応答する。この受信されたフレームデータは、(フレームピースが送信されたとき)既存のフレームに組み込まれても良く、フルフレームとして組み込まれてもよい。10

#### 【0039】

このように、図2の方法200で説明したように、ビデオフレーム情報を用いて、低レベルのパケットロスを検出した時にどうするか判断し、かかる倍には、データが失われた時には、復号レイテンシ及び視覚的破損を最小化する。実施形態は、ネットワークにより入来する低レベルデータパケットを処理し、そのデータパケットをフレーム及びフレームピースによりトラッキング及びグループ分けし、他のフレームデータパケットが到着するのを待っている間に、グループ分けしたフレームピースを復号モジュール105に供給する。フレームピースのデータパケットがエラー訂正により回復できないとき、部分的フレームピースは、リフレッシュが行える(すなわち、フルフレーム111が送信される)まで、視覚的破損(*visual corruption*)を最小化する合成データにより、完成又は置換される。これによりさらに、有効データを供給することによりロバストなエラー隠蔽機能を有しないビデオデコーダの使用が可能となる。20

#### 【0040】

実際、幾つかの実施形態では、コンピュータシステム101は、受信したフレームピース110の少なくとも1つが、フォワードエラーコレクションデータ(*forward error correction data*)を含むと判断してもよい。この前方エラー訂正(FEC)データを実施して、フレームピース中のエラーを修正し、最終的にそのフレームピースを用いてフレームを構成してもよい。かかる場合、フレームピースは、前方エラー訂正が開始される前に復号されてもよい。他の実施形態では、コンピュータシステム101は、完全なフレームピースの復号を続ける間に、受信されていないフレームピースを有するフレームのFEC情報を待つように構成されていてもよい。このように、復号モジュール105は、(データパケットが失われ又は破損することにより)不完全な又は正しくないデータを有するフレームピースのFECデータを待つ間でも、フレームピースの復号を続けても良い。次いで、FEC情報が対応するフレームピースのFECデータが受信されたとき、復号モジュール105は、FEC補正フレームピースを復号して、そのフレームピースを他のフレームピースと組み合わせて、復号フレーム109を形成してもよい。このフレームはディスプレイ108に表示されてもよい。30

#### 【0041】

図7は、H.264フォーマットを実装する一実施形態を示す。このフォーマットが使われるが、言うまでもなく、同様の手法で他のフォーマットを用いてもよい。フレームデータパケットは複数のフレームにわたりトラッキングされる。これにより、例えば、より前のフレームが完了されるべきであり、失われたパケットは到着せず、もう無いと考えるべきであることを、コンピュータシステムに表示するトラッキング情報が得られる。図7のH.264チャネル702はフレームトラッカーオブジェクト703A、703B及び703Cを用い、フレームに関連するデータパケットと共に、現在進行中の既知のフレームをトラッキングする。これらのデータパケットは、ビデオ出力チャネル701又はその他のストリーミング源(例えば、リアルタイムトранSPORTプロトコル(RTP)ソース)から受信される。フレームピース(又は、ここでは「スライス」)は、各フレームの進行中のスライスの各々をトラッキングするスライストラッカーオブジェクト704A、4050

704B 及び 704C によりトラッキングされる。これにより、コンピュータシステム 101 は、新しいフレームがいつ始まるか、前のフレームはどのくらい完全であったか、どのスライスが完全又は失われたが、フレームはいつ前方エラー訂正を必要とするかということなどを知ることができる。

#### 【0042】

H.264 チャネル 702 は、前処理無しに、ビデオ出力チャネル 701 から原 RTP データパケット（又は他のプロトコルのデータパケット）を受信してもよい。タイムスタンプに基づき、フレームデータパケットは、すでにあるものが、又は新しく生成されたもの（例えば、703C）が、既存のフレームトラッカーオブジェクト（例えば、703A）に送られる。フレームデータパケットペイロードは、フレームトラッカーにより受信された時に分析され、スライスネットワーク抽象レイヤユニット（NALU）の開始を含む場合、新しいスライストラッカー（例えば、704C）が生成され、リストに追加される。フレームデータパケット（例えば、RTP パケット 708）は、既存のスライスのシーケンス中の次の物であるとき、対応するスライストラッカーに送られる。そうでなければ、フレームデータパケットは、アウトオブオーダー（OOO）パケットリスト（例えば、706）に保持されるか、もし複製であれば無視される。パケットは、前方エラー訂正（FEC）データ（例えば、707）を含むとき、無くなつたパケット（場合によっては、失われるパケットは 1 つだけである）の再生にすぐに使われるか、又はより多くのパケットが到着していれば、後で使う為、フレームトラッカーにより保持される。

#### 【0043】

少なくとも幾つかの実施形態では、フレームデータパケットは、スライストラッカー（例えば、704A）に送られた場合、正しい順序であると仮定することができる。スライスの最後のフレームデータパケットが受信されると、スライス NALU（例えば、705）は、ビデオデコーダ（例えば、図 1 の 105）に送られ得る。

#### 【0044】

あるフレーム 111 のスライス（例えば、フレームピース 110）が完成され、デコーダ（例えば、105）に送信された場合、そのフレームは完成したものとみなされ（例えば、705）、チャネルがそのフレームをディスプレイ 108 に表示するため転送することができる。次のフレームのデータが到着し始めた時に、フレーム及びその一以上のスライスが完成していないとき、コンピュータシステム 101 は、エラー訂正により再生できないフレームデータは、無くなつたと判断してもよい。かかる場合には、不完全なフレームデータにより生じるエラーを隠すために、フレームピース合成が行われる。失われたフレームピースは、フレーム合成モジュール 107 により合成され、復号モジュール 105 がフレームの処理を終了できるようにする。図 7 で説明した H.264 実施形態では、失われた各スライスに対して、前の参照フレームのピクセルをコピーした合成スライス NALU が構成される。合成スライス NALU は復号のためデコーダ 105 に送られる。その後の完全なスライスも、フレームを形成するために送られる。これにより、フレームデータが不完全又は破損している場合に、完全な出力フレームを生じる。

#### 【0045】

破損したフレーム（すなわち、合成部分 109 を有する復号フレーム）は、その後のフレームからは参照されない場合、ディスプレイ 108 に表示でき、次のフレームは正常に処理される。フレームデータ破損は見えるが、1 つのフレーム時間のみであり、後のフレームには伝搬しない。または、破損その他のデータ損失が十分大きければ、フレームは完全にドロップされ、まったく表示されないこともできる。破損フレームが参照フレーム（例えば、H.264 フォーマットの P フレーム）である場合、その破損はその後のフレームに伝搬する（実際、その後のフレームの基礎として用いられるフレームは、その破損は伝搬する）。かかる場合には、破損フレームは表示されるが、ビデオソースは、データロスが発生したことを、ビデオ出力チャネル 701 を介して通知される。これにより、ビデオプロバイダ（例えば、図 1 のコンピュータシステム 113）は、（幾つかの場合には、次の又はその次で）破損のフレッシュかつ素早いクリーンアップをすぐにトリガーする。

10

20

30

40

50

**【 0 0 4 6 】**

ここで図3に戻り、フローチャートは、不完全なビデオデータを復号する方法300を示す。環境100のコンポーネント及びデータを頻繁に参照して、方法300をここで説明する。

**【 0 0 4 7 】**

方法300は、一以上のフレームピースの受信に応じて、少なくとも1つの追加フレームピースを受信するのを待っている間に、受信した一以上のピースを復号する動作(動作310)を含む。例えば、コンピュータシステム101の通信モジュール104Aは、フレームピース110を受信してもよい。これは、結合されると、フルフレーム111を構成する。追加フレームピースが受信されるのを待っている間、復号モジュール105は、受信したフレームピースが順序通りでないとしても、受信したフレームピースの復号を開始してもよい。状態モニタリングモジュール106は、少なくとも1つの追加フレームがまだ受信されていないという点でフレームが不完全であることを示す状態の発生を検出してもよい(動作320)。この状態は、フレームピースのフレームデータがまだ受信されていない、又はフルフレームピースが受信されていないとの判断であり得る。追加的に又は代替的に、この状態は、後続の少なくとも1つのフレームピースを受信すること、又はあるピースのフレームデータが破損しているか、そうでなければ使えないと判断することであってもよい。

**【 0 0 4 8 】**

この検出動作に応じて、フレーム合成モジュール107は、まだ受信していない追加フレームピースを置き換えるフレームピースを合成してもよい(動作330)。合成は様々な異なる方法で行われても良いが、各方法は、失われた又は破損したデータの合成又は再生を行うことになる。幾つかの場合には、フレームピースを合成して受信されていないフレームピースを置き換えることは、前のフレームに関連するフレームデータを用いてフレームピースを合成することを含む。かかる場合には、フレーム合成モジュールは、以前の完全な一以上のフレームにアクセスし、分析して、現在のフレームが失われ又は破損しているフレームデータの部分のピクセルがどう見えるか判断してもよい。この情報を用いて、フレーム合成モジュール107は、前のフレームに現れたフレームデータと類似した、又は同一のフレームデータを生成してもよい。

**【 0 0 4 9 】**

さらに、フレーム合成モジュール107は、フレーム間の前の動きを決定する動作を含む様々なファクタを考慮してもよい。フレーム間の前の動きは、データが、合成されるフレームにおいてどのように変化し続けるかのインジケータであってもよい。さらに、フレーム合成モジュール107は、一以上の前のフレームに基づいて、カメラがどの方向にパンしているか判断してもよく、決定されたパン方向を用いてフレームピースを合成してもよい。よって、カメラが左から右にパンしていると判断されると、フレーム合成モジュール107は、フレームの右側にあったピクセルが、そのフレームの左側に徐々に近づく動きをすると判断する。動きベクトルを含む他のファクタを用いて、(除々に左から右へ又は上から下へのパンとは異なり)不規則な動きを決定し、現在のフレーム(すなわち、合成されるフレーム)がどう見えるべきかに関するその他のインジケータを提供してもよい。この情報を用いて、フレーム合成モジュール107は、フレームピースの失われたフレームデータを近似又は合成できる。これは前に復号したフレームピースと合成されて、完全なフレーム111を構成する。

**【 0 0 5 0 】**

図4は、不完全なビデオデータを復号する方法の別の一例(方法400)を示すフローチャートである。環境100のコンポーネント及びデータを頻繁に参照して、方法400をここで説明する。

**【 0 0 5 1 】**

方法400は少なくとも1つのフレームピースにアクセスする動作を含み、フレームは一以上のフレームピースを含み、各フレームピースは個別に復号可能である(動作410)

10

20

30

40

50

)。復号モジュール 105 は、少なくとも 1 つの追加フレームピースが受信されるのを待つ間に、一以上の受信フレームピース 110 にアクセスし、これらのフレームピースを復号する（動作 420）。状態モニタリングモジュール 106 は、少なくとも 1 つの追加フレームピースがまだ受信されていないと判断し、フレームを不完全なままレンダリングする（動作 430）。次いで、判断に応じて、フレーム合成モジュールは、少なくとも 1 つの前のフレームに関連するフレームデータを用いて、フレームピースを合成する。フレームとフレームピースは、図 7 に概略を示したように、別々にトラッキングされてもよい。複数のフレームにわたるフレームピースデータパケットをトラッキングすることにより、コンピュータシステム 10 は、後続のフレームのデータパケットがいつ受信されるかを判断してもよい。後続フレームのデータパケットが受信されるとき、フレーム合成を開始させる表示又は状態であってもよい。10

#### 【0052】

幾つかの実施形態では、フレームピーストラッキング及び／又はフレームトラッキングは、プロトコルベースシーケンス番号を用いて、インクリメント状に増加するタイムスタンプを用いて、又はその他のタイプの識別子を用いて、実装してもよい。プロトコルベースシーケンス番号又はインクリメント状に増加するタイムスタンプを、各フレーム及び／又は各フレームピースに割り当ててもよい。これにより、フレーム又はフレームピースを個別にトラッキングでき、その後、新しいフレームピース又はフレームのフレームデータを受信した時に、コンピュータシステム 101 にアラートを発することができる。例えば、フレーム及び／又はフレームピースは、ストリーミングされたビデオゲームの一部として受信されてもよい。ストリーミングされたビデオゲームは、一般的に、高いフレームレートで実行され、ユーザが遅れを感じないようにされる。かかるシナリオでは、コンピュータシステム 101 は、プロトコルベースシーケンス番号又はインクリメント状に増加するタイムスタンプを用いて、フレーム又はフレームピースが不完全であるか、及びそのフレームの部分を合成する必要があるか、素早く判断してもよい。不完全である（例えば、フレームピースデータパケットをフレームピース及び／又はフレームと関連付けるのに使われるタイムスタンプ情報が、新しいフレームが受信されることを示す）とき、フレーム合成モジュールは、失われた又は破損したフレームデータをすぐに合成できる。このように、フレームはすばやく完成でき、ゲームプレーヤに表示するため、ディスプレイ 108 に送信できる。20

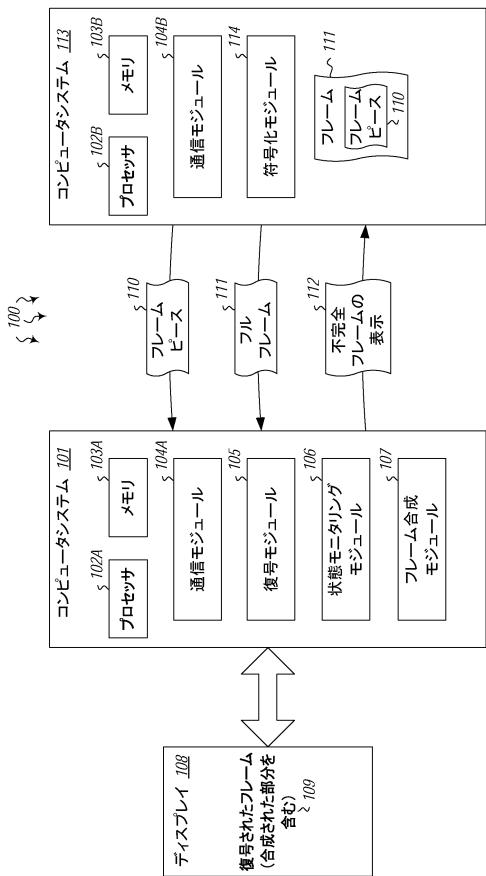
#### 【0053】

したがって、不完全なビデオデータを復号する方法、システム及びコンピュータプログラム製品を提供する。フレームピースは、フレームの他のピースを待っている間に、復号され得る。さらに、フレームデータのいずれかの部分が失われた又は破損したと判断されたとき、そのフレームのその部分を、その失われた又は破損したデータの見え方を近似するように、合成できる。30

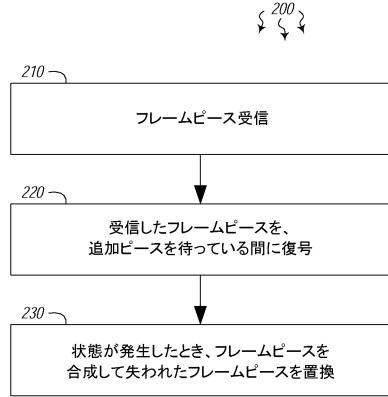
#### 【0054】

ここに説明のコンセプトとフィーチャ（features）は、その精神や説明した特徴から逸脱することなく、他の具体的な形式で実施してもよい。説明の実施形態は、すべての点で、例示であり限定ではないと考えなければならない。それゆえ、開示の範囲は、上記の説明ではなく、添付の特許請求の範囲により示される。請求項の意味又は均等の範囲内のすべての変更は、その範囲内に含まれる。40

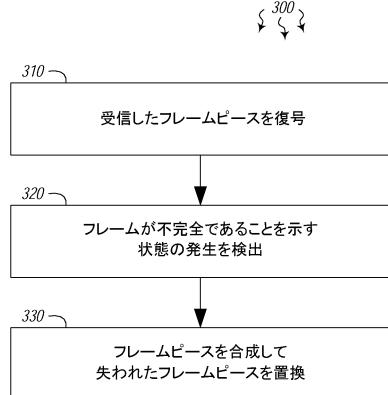
【図1】



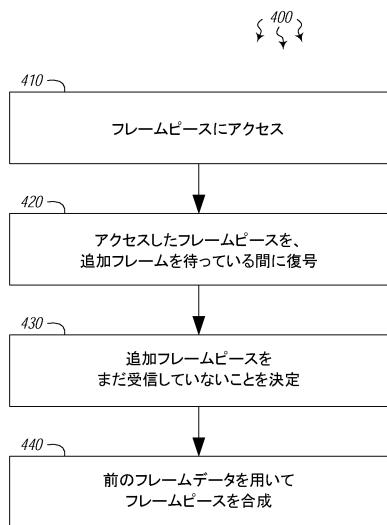
【図2】



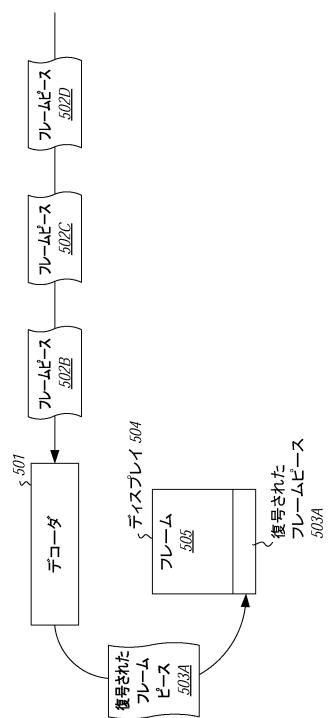
【図3】



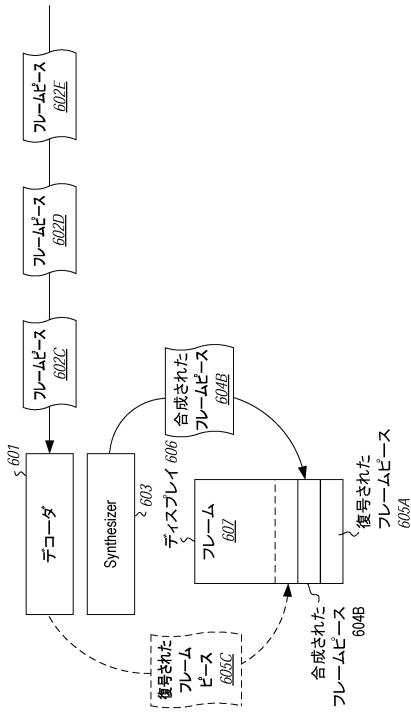
【図4】



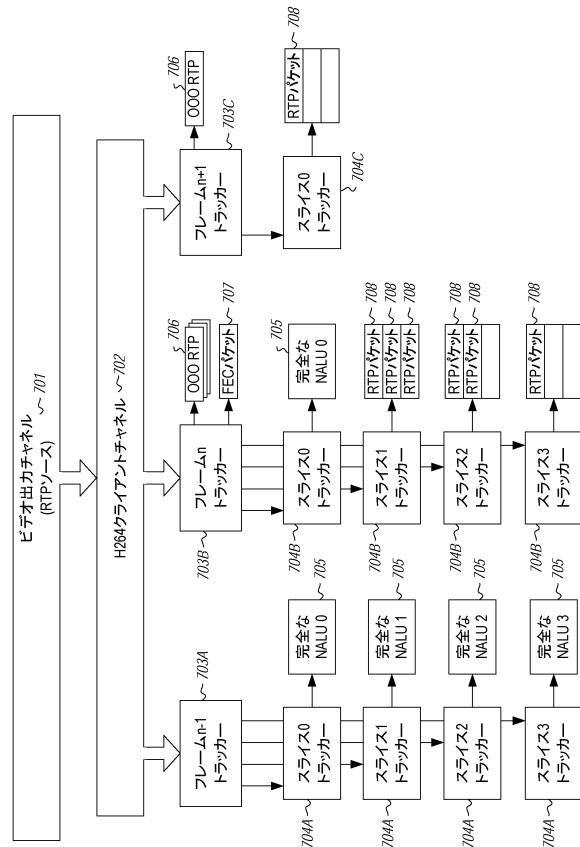
【図5】



【図6】



【図7】



---

フロントページの続き

(72)発明者 キャラハン , ショーン

アメリカ合衆国 98052-6399 ワシントン州 レッドモンド ワン マイクロソフト  
ウェイ マイクロソフト テクノロジー ライセンシング , エルエルシー エルシーエー - インタ  
ーナショナル パテンツ (8/1172) 内

審査官 久保 光宏

(56)参考文献 特開平10-13248(JP,A)

特開平10-145237(JP,A)

特表2011-507406(JP,A)

特開平7-322248(JP,A)

特開2008-312125(JP,A)

特開2009-65259(JP,A)

特開2005-175787(JP,A)

特開平8-46599(JP,A)

特開2002-290381(JP,A)

Yao Wang, et al., "Error Control and Concealment for Video Communication: A Review", Proceedings of the IEEE, 1998年 5月 1日, Vol.86, No.5, Pages 974-997, ISSN:0018-9219

大久保 榮 監修 , 「改訂三版H.264/AVC教科書」 , 日本 , 株式会社インプレスR & D , 2009年 1月 1日 , 第1版 , 第200 ~ 201頁 , ISBN:978-4-8443-2664-9

渡辺 裕 , 「MPEGビデオ符号化方式」 , 電子情報通信学会技術研究報告 , 日本 , 社団法人電子情報通信学会 , 2009年 4月 9日 , Vol.109, No.3 , 第71 ~ 76頁 , ISSN:0913-5685

平島章史 (外1名) , 「音声・ビデオIP伝送のためのQoSベースビデオ出力方式における閾値設定法の比較」 , 電子情報通信学会技術研究報告 , 日本 , 社団法人電子情報通信学会 , 2010年 1月 14日 , Vol.109, No.376 , 第201 ~ 206頁 , ISSN:0913-5685

(58)調査した分野(Int.Cl. , DB名)

H04N19/00-19/98

H04N21/00-21/858

CSDB (日本国特許庁)

IEEE Explorer (IEEE)