

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4959504号
(P4959504)

(45) 発行日 平成24年6月27日(2012.6.27)

(24) 登録日 平成24年3月30日(2012.3.30)

(51) Int.Cl. F I
HO 4 N 7/173 (2011.01) HO 4 N 7/173 6 3 0
HO 4 N 7/26 (2006.01) HO 4 N 7/13 Z

請求項の数 9 (全 20 頁)

(21) 出願番号	特願2007-276004 (P2007-276004)	(73) 特許権者	390035493
(22) 出願日	平成19年10月24日(2007.10.24)		エイ・ティ・アンド・ティ・コーポレーション
(62) 分割の表示	特願平10-543006の分割		AT&T CORP.
原出願日	平成10年4月7日(1998.4.7)		アメリカ合衆国 10013-2412
(65) 公開番号	特開2008-118637 (P2008-118637A)		ニューヨーク ニューヨーク アヴェニュー
(43) 公開日	平成20年5月22日(2008.5.22)		ー オブ ジ アメリカズ 32
審査請求日	平成19年11月1日(2007.11.1)		
審判番号	不服2010-23704 (P2010-23704/J1)		
審判請求日	平成22年10月21日(2010.10.21)		
(31) 優先権主張番号	60/042, 798		
(32) 優先日	平成9年4月7日(1997.4.7)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 適応制御を行うことができるMPEGコード化オーディオ・ビジュアルオブジェクトをインターフェイスで連結するためのシステムおよび方法

(57) 【特許請求の範囲】

【請求項1】

Java（登録商標）アプリケーションと、MPEGコード化メディアを復号及び出力するメディア出力デバイスとの間で相互に作用可能な通信を可能とするシステムであって、該システムが、

プレゼンテーションエンジンを備え、該プレゼンテーションエンジンが、MPEG-4ビットストリームを受信するためのデジタルメディア集積フレームワーク(DMIF)、該MPEGビットストリームを、シーン記述情報を含む第1の部分とオーディオ・ビジュアルメディアオブジェクトを含む第2の部分に分離するためのDemux、該Demuxからの該第2の部分をバッファリングするための少なくとも1つの第1のバッファ、該第1のバッファからのメディアを復号するための少なくとも1つのメディアデコーダ、該復号されたメディアをバッファリングするための少なくとも1つの第2のバッファ、該第2のバッファからの復号されたメディアを相互に配列し、シーンを合成するコンポジタ、該合成されたシーンをレンダリングしてディスプレイに送信するためのレンダリング装置、及び前記Demuxからの前記第1の部分に基づいて該メディアデコーダ、前記コンポジタ及び前記レンダリング装置を制御するBIFSデコーダを備え、

前記システムがさらに、

アプリケーションエンジンを備え、該アプリケーションエンジンが、

Java（登録商標）仮想機械(JVM)(110)、

JVM(110)に接続されたMPEGアプリケーション(100)、及び

受信 M P E G - 4 ビットストリームのメディアを復号し、復号された該メディアからシーンを合成してレンダリングし、該シーンをディスプレイに送るための前記プレゼンテーションエンジンに対して該アプリケーションエンジンがどのようにして通信するかを定義する複数の A P I

からなり、

前記複数の A P I が、ユーザ入力又は前記メディア出力デバイスに利用可能なシステム資源に応じた前記プレゼンテーションエンジンの制御を可能にする、システム。

【請求項 2】

請求項 1 のシステムにおいて、前記複数の A P I が、オーディオ・ビジュアルオブジェクトを復号するメディアデコーダにアクセスして制御する少なくとも 1 つのメディアデコーダ A P I を含む、システム。

10

【請求項 3】

請求項 2 のシステムにおいて、該少なくとも 1 つのメディアデコーダ A P I がデコーダを開始、停止及び再開させるシステム。

【請求項 4】

請求項 2 のシステムにおいて、該システムがさらに、該 M P E G ビットストリームを復号するための準備において少なくとも 1 つのメディアデコーダ (2 7 0) のうちの 1 つのメディアデコーダを M P E G ビットストリームに接続し、該 M P E G ビットストリームを該メディアデコーダから送出するモジュールを備えるシステム。

【請求項 5】

20

複数のアプリケーション・プログラミング・インターフェイス (A P I) を介した、J a v a (登録商標) 言語を用いる M P E G アプリケーションとメディア出力デバイスの間の通信に基づいて、M P E G ビットストリームを復号及びレンダリングする方法であって、

M P E G コード化メディアを復号及び出力する前記メディア出力デバイスと前記 M P E G アプリケーション (1 0 0) との相互作用性を制御することに関連する基本ビットストリームを受信するステップ、及び

該 M P E G アプリケーション (1 0 0) とプレゼンテーションエンジンとの間で前記複数のアプリケーション・プログラミング・インターフェイス (A P I) を介して通信するステップであって、前記プレゼンテーションエンジンは、M P E G - 4 ビットストリームを受信するデジタルメディア集積フレームワーク (D M I F)、該 M P E G ビットストリームをシーン叙述情報を含む第 1 の部分とオーディオ・ビジュアルメディアオブジェクトを含む第 2 の部分に分離する D e m u x (1 6 5)、該 D e m u x からの該第 2 の部分をバッファリングする少なくとも 1 つの第 1 のバッファ (2 5 1)、該第 1 のバッファからのメディアを復号する少なくとも 1 つのメディアデコーダ (2 7 0)、復号された該メディアをバッファリングする少なくとも 1 つの第 2 のバッファ (2 7 6)、該第 2 のバッファからの復号メディアを相互に配列してシーンを合成するコンポジタ (2 8 2)、合成した該シーンをレンダリングしてディスプレイに送るレンダリング装置 (2 8 9)、及び前記 D e m u x からの該第 1 の部分に基づいて前記メディアデコーダ、前記コンポジタ及び前記レンダリング装置を制御する B I F S デコーダ (2 2 5) を備え、前記複数の A P I が、ユーザ入力又は前記メディア出力デバイスに利用可能なシステム資源に応じた前記プレゼンテーションエンジンの制御を可能にする、ステップ

30

からなる方法。

【請求項 6】

請求項 5 の方法であって、さらに、

前記 M P E G アプリケーションにおいてユーザ入力を受信するステップ、及び

該複数の A P I のオーサリング機能の使用を介して、該ユーザ入力に基づいて、該メディア出力デバイスによってレンダリングされる該 M P E G コード化メディアの該メディアデバイス出力の一部としてシーンを修正するステップ
を備える方法。

40

50

【請求項 7】

請求項 5 の方法において、前記複数の A P I が、デコーダを開始、停止及び再開させるメディアデコーダ A P I (2 2 0) を含む、方法。

【請求項 8】

請求項 7 の方法において、該メディアデコーダ A P I (2 2 0) がさらに、該 M P E G ビットストリームを復号するための準備において少なくとも 1 つのメディアデコーダ (2 7 0) のうちの 1 つのメディアデコーダを M P E G ビットストリームに接続し、該 M P E G ビットストリームを該メディアデコーダから送出するモジュールを備える、方法。

【請求項 9】

請求項 5 の方法において、前記複数の A P I が、前記オーディオ・ビジュアルメディアオブジェクト及び前記 B I F S シーン叙述を含む M P E G - 4 ビットストリームの編集及び修正を助けるためのストリーム編集 A P I (5 0 1) を含み、該ストリーム編集 A P I によって前記 M P E G アプリケーション (1 0 0) にシーングラフ (2 2 5) を操作するためのプログラミング・インターフェイスを提供するシーングラフ A P I (2 1 0) を含む、方法。

10

【発明の詳細な説明】

【技術分野】

【0001】

本出願は、優先権を主張する米国特許仮出願 6 0 / 0 4 2 , 7 9 8 に関連する。

本発明は、コード (符号) 化マルチメディアおよびその記憶、ユーザへの配布に関し、特に、チャンネルおよび復号資源のどちらかが制限され、時間により変動する場合、またはユーザ・アプリケーションが、符号化されたマルチメディアオブジェクトとの高度の相互作用を必要とする場合の符号化に関する。

20

【背景技術】

【0002】

デジタルマルチメディアは、操作面、複数のものを発生できるという面およびエラーに影響されないという利点およびその他の利点を持っているが、送信帯域幅の記憶容量により制限を受ける。それ故、実際に使用する場合には、多くの場合、圧縮または符号化する必要がある。さらに、インターネットおよび他のネットワーク上の、デジタルマルチメディアに対する需要が急速に増大しているため、効率的な記憶装置、ネットワークアクセスおよびサーチおよび検索、多くの符号化スキーム、記憶フォーマット、検索技術および送信プロトコルが開発されてきた。例えば、画像およびグラフィックス・ファイル用には、G I F、T I F および他のフォーマットが使用されている。同様に、オーディオ・ファイルも、R e a l A u d i o、W A V、M I D I および他のフォーマットにより符号化され、記憶される。アニメーションおよびビデオ・ファイルは、多くの場合、G I F 8 9 a、C i n e p a k、I n d e o および他のフォーマットにより記憶される。

30

【0003】

非常に多数の現存のフォーマットを再生するために、多くの場合、デコーダおよびインタープリタが必要になるが、その場合、前記デコーダおよびインタープリタが、ハードウェアで実行されるのか、ソフトウェアで実行されるのかにより、特にソフトウェアの場合には、ホストコンピュータの能力により、速度、品質および性能がまちまちになる。マルチメディアの内容が、コンピュータ (例えば、パソコンのような) を通してアクセスされた、ウェブページ内に含まれている場合には、ウェブブラウザを予想されるすべての内容に対して正しく設定する必要があり、また前記内容を処理するために、各タイプの内容を認識し、内容ハンドラ (ソフトウェア・プラグインまたはハードウェア) の機構をサポートしなければならない。

40

【0004】

相互動作が可能でなければならないこと、品質および性能が保証されていなければならないこと、チップ設計の際のスケールが小さくならないこと、また種々様々のフォーマットのための内容を発生する際にコストがかかること等の理由で、マルチメディア

50

コード化、パケット化、および確実な配布の分野で標準化が促進されてきた。特に、I S O M P E G (国際規格組織動画専門家グループ)は、M P E G - 1およびM P E G - 2と呼ばれる、二つの規格の形の符号化マルチメディアに対するビットストリーム・シンタックスおよび復号意味論の標準化を進めてきた。M P E G - 1は、主として、コンパクトディスク(C D)のようなデジタル記憶媒体(D S M)上で使用するためのものであり、一方、M P E G - 2は、主として、放送環境(トランスポート・ストリーム)で使用するためのものである。しかし、M P E G - 2は、またD S M(プログラム・ストリーム)上で使用するための、M P E G - 1類似の機構もサポートする。M P E G - 2は、また独立のまたはネットワークに接続している、M P E G - 2の標準化再生のために必要な場合がある、基本的なユーザの双方向通信のためのD S Mコマンドおよび制御のような追加機能を含む。

10

【 0 0 0 5 】

安価なボード/P C M C I Aカードが開発され、中央処理装置(C P U)が入手することができるようになったために、M P E G - 1規格が、パソコン上の映画およびゲームの再生に共通に使用することができるようになった。一方、M P E G - 2規格は、比較的高画質のアプリケーション用のものであるため、デジタル衛星T V、デジタルケーブルおよびデジタル汎用ディスク(D V D)からの、娯楽用アプリケーションに共通に使用されている。前記アプリケーション/プラットフォームの他に、M P E G - 1およびM P E G - 2は、ネットワークを通して送られるストリーム、ハードディスク/C D上に記憶されているストリーム、およびネットワークによるアクセスおよびローカル・アクセスの組み合わせの種々の他のコンフィギュレーションでの使用が期待されている。

20

【 0 0 0 6 】

M P E G - 1およびM P E G - 2の成功、インターネットおよび移動チャネルの帯域幅制限、ブラウザを使用するウェブをベースとするデータアクセスの柔軟性、および双方向個人通信の必要が増大しているために、マルチメディア使用および制御のための新しいパラダイムが誕生した。それに応じて、I S O M P E Gは、新しい規格、M P E G - 4を制定に着手した。前記M P E G - 4規格は、個々のオブジェクトの形のオーディオ・ビジュアル情報の符号化、および前記オブジェクトの合成および同期再生用のシステム向けのものである。前記固定パラメトリックシステムのM P E G - 4の開発が、引き続き行われる一方で、通信、ソフトウェアおよびJ a v a (登録商標)言語によるようなネットワーク運営用の新しいパラダイムにより、柔軟で、適応性を持ち、ユーザが双方向で通信できる新しい機会が生まれた。

30

【 0 0 0 7 】

例えば、J a v a (登録商標)言語の開発により、ユーザがアクセスしたウェブページで、ホストとなるウェブサーバからの、クライアントパソコン上での、アプレット(ジャバ・クラス)のダウンロードおよび実行に重要な、ネットワーク運営およびプラットフォームの独立が可能になる。アプレットの設計により、サーバ側で記憶しているデータへの一回だけのアクセスが必要になる場合もあるし、すべての必要なデータを、クライアントのパソコン上に記憶することもできるし、または(スタートアップに必要な記憶スペースを少なくし、時間を短縮するために)数回の部分的アクセスが必要になる場合がある。後のシナリオは、ストリーム型再生と呼ばれる。

40

【 0 0 0 8 】

すでに説明したように、符号化マルチメディアを、例えば、パソコンのようなコンピュータ上で、インターネットおよびローカルのネットワーク接続アプリケーションのために使用し場合には、多くの状況が発生する。第一に、マルチメディアのネットワークによるアクセス用の帯域幅は、制限されるか、時間と共に変化するようになり、その場合には、最も重要な情報だけを送信し、その後で、もっと広い帯域幅を使用できるようになったとき、他の情報を送信しなければならない。

【 0 0 0 9 】

第二に、使用できる帯域幅がどのような幅のものであっても、復号を行わなければなら

50

ないクライアント側のパソコンのCPUおよび/またはメモリ資源は制限を受け、さらに、前記資源も時間により変化する。第三に、マルチメディア・ユーザ（消費者）は、高度に双方向の非直線ブラウジングおよび再生を必要とする場合がある。これは特別のことではない。何故なら、ウェブページ上の多くのテキスト内容は、ハイパーリンクされた機能を使用することにより、ブラウズすることができ、符号化オーディオ・ビジュアルオブジェクトを使用する表示用に、同じパラダイムの使用が期待されるからである。パラメトリックMPEG-4システムは、非常に制限された方法でしか、前記の状況処理できないだろう。すなわち、オブジェクトの欠落またはオブジェクトの一時的な発生が起こり、復号または表示ができなくなり、その結果、オーディオ・ビジュアルの表示が不安定になる。

10

【0010】

さらに、MPEG-4を使用した場合、ユーザは、前記状況をうまく処理することができない。前記パラメトリックシステムの前記制限を回避するの、MPEG-4開発用の可能性のある一つの選択肢は、プログラムを使用するシステムの開発である。

【0011】

ソフトウェア業界では、長い間、アプリケーション・プログラミング・インターフェイス（API）が、多数の異なるタイプのコンピュータ・プラットフォーム上での標準化動作および機能を達成するための手段として認識されてきた。通常、APIを定義することにより、種々の動作を標準化することができるが、前記動作の性能は、特定のプラットフォームに関心を持つ特定のベンダが、そのプラットフォーム用に最適化された実行を供給する場合があるので、種々のプラットフォーム上で異なる場合がある。グラフィックの分野では、バーチャルリアリティモデリング言語（VRML）は、シーングラフ・アプローチを使用することにより、オブジェクトとシーンの説明との間の、空間的および時間的關係を指定する手段を使用することができる。MPEG-4は、リアルタイムのオーディオ/ビデオ・データ、および顔または体の動きを処理するために、多くの方法で、VRMLおよび拡張VRMLに対して、中心の構造物の二進表示（BIFS）を使用してきた。VRMLの機能を強化し、プログラムによる制御を行うことができるように、ディメンションX社は、一組のAPIを液体リアリティとして発売した。最近、サン・マイクロシステム社が、ジャバ3Dの初期バージョン、すなわち、他の機能も持っているが、とりわけ、合成オーディオ・ビジュアルオブジェクトの、シーングラフとしての表示をサポートするAPI仕様を発表した。サン・マイクロシステム社は、ジャバ・メディア・フレームワーク・プレーヤAPI、すなわち、マルチメディア再生用のフレームワークを発売した。しかし、現在市販されているAPIパッケージは、どれもMPEG-4符号化および他の高度のマルチメディアの内容の種々の要求に適した広範でしっかりした機能を持っていない。

20

30

【発明の開示】**【課題を解決するための手段】****【0012】**

本発明は、パラメトリックMPEG-4のような非適応クライアントシステムが適応できるような方法で、符号化されたオーディオ・ビジュアルオブジェクトを再生およびブラウズすることができるようにする、符号化オーディオ・ビジュアルオブジェクトをインターフェイスにより接続するためのシステムおよび方法を提供する。本発明のシステムおよび方法は、アーキテクチャレベルでプログラムを使用するもので、特に、MPEG-4符号化データにアクセスし、それを処理するために構成された、指定の組のアプリケーション・プログラミング・インターフェイスにより、パラメトリックシステムが一番上に適応性のある一つの層を追加する。

40

【0013】

当業者には周知のMPEG-4は、デジタルメディア集積フレームワーク（DMIF）が監督するシステムズ・デマルチプレクス（Demux）、シーングラフ、およびメディアデコーダ、バッファ、コンポジットおよびレンダリング装置からなるパラメトリックシステムであると見なすことができる。標準MPEG-4に対して本発明のシステムおよび方

50

法が行った強化または拡張としては、メディア復号の範囲に入る一組の指定API、ユーザ機能およびクライアントアプリケーションが内蔵することができるオーサリング等がある。強力なオーディオ・ビジュアル・インターフェイス装置を提供することにより、ユーザの入力に応じて多数の強化リアルタイムおよび他の機能を実現することができ、MPEG-4クライアントが使用することができるシステム資源が少ない場合に、滑らかなデグラデーションを行うことができる。

【0014】

本発明が行われた一つの理由は、ユーザの制御の下でMPEG-4の再生およびブラウズを行うための標準化インターフェイスが、必要であったためである。本発明のインターフェイスは、符号化メディアデータを、直ちに使用することができるターミナル資源に容易に適應させるためのものである。本発明のインターフェイスは、また機能として直接、または近い将来に重要になるとと思われるオーディオ・ビジュアル・アプリケーションおよびサービスに埋設して、間接的に、ユーザが求めていると思われる相互作用を容易にする。

10

【0015】

本発明は、いくつかの分野を含む、しっかりしたアプリケーション・プログラミング・インターフェイス(API)の形の、インターフェイスによる接続方法を指定する。メディア復号の分野においては、ビジュアル復号インターフェイスが指定される。ユーザ機能の分野においては、順送り、ホットオブジェクト、方向性、トリックモードおよび透明なインターフェイスが指定される。ユーザ・オーサリングの分野においては、ストリーム編集インターフェイスが指定される。インターフェイスの前記全体の組は、すべての組を網羅しているわけではないが適應性のかなりの部分の実行を容易にする。

20

【0016】

添付の図面を参照しながら、本発明を説明する。図面中、類似の素子には類似の番号が使用されている。

【発明を実施するための最良の形態】

【0017】

本発明のシステムおよび方法を、MPEG-4復号環境で説明する。前記環境において、本発明は、一つのAPIではなく、種々のインターフェイスを拡張MPEG-4システムのためのAPIの集合体を指定する。当業者には周知のJava(登録商標)言語が、APIを指定するために使用され、また当業者には周知の関連電子メモリ、記憶装置、バスおよび関連構成部分を持つ汎用プロセッサまたは特殊用途プロセッサ上で実行される。本発明においては、例示として、三つの分野のAPIが識別され、各分野において、代表的な機能が提供される。

30

【0018】

上記三つの例示としてのAPIは下記の通りである。

- ・メディア復号
- ・ユーザ機能
- ・オーサリング

本発明の特定のAPIおよび前記APIの実行を組織する方法を、下記表に最初まとめ、以下に説明する。

40

表1 本発明のAPI

【表 1】

No	A P I 分類および指 定	説明	
1.	メディア解読 ビジュアル解読	コード化ビットストリームからの、スケーラビリティを持つまたは持たないビジュアル対象物の解読	10
2.	機能 順送り	ユーザ制御の下での A V 対象物の順送り解読および合成	20
3.	ホットオブジェクト		
4.	方向性	ユーザの制御に基づく A V 対象物の解読、強化および合成	
5.	トリック・モード	ユーザが選択した視点（または音響）方向への A V 対象物の解読	
6.	透明	ユーザの制御の下での、A V 対象物の一部の解読および A V 対象物の合成 透明およびユーザ制御に基づく A V 対象物の解読 精製および合成	30
7.	部分オーサリング ストリーム編集	解読および再度のコード化を行わないで内容を修正するための、MPEG-4 ビットストリームの編集	40

【0019】

パッケージは、A P I の実行を組織するための手段である。本発明が表示する A P I のライブラリを考慮にいれた、パッケージの部分的なリストは下記の通りである。

- ・ m p g j . d e c

このパッケージは、相互作用を含むユーザ機能に対するクラスを含む。

- ・ m p g j . f u n c

このパッケージは、優先を含むユーザ機能に対するクラスを含む。

- ・ m p g j . u t i l

このパッケージは、種々の入力、出力、サウンドおよびビデオ装置へのインターフェイスを提供する。

【 0 0 2 0 】

ここで、本発明のシステムおよび方法および関連インターフェイス方法（API）について説明する。

図1は、本発明のシステム実行のハイレベルのブロック図である。前記実行は、二つの主要な構成部分からなる。第一の構成部分は、デジタルメディア集積フレームワーク（DMIF）160からなる周知のパラメトリックシステムで、チャンネル155に送付インターフェイスを供給し、システムズ・デマックス165に接続している。前記システムの出力は、図を簡単にするため、BIFSおよびメディアデコーダ、バッファ、コンポジットおよびレンダリング装置170からなる一つの集合ブロックで示す、一連のブロックを通して送られる。レベル175上の前記集合ブロックの出力は、ディスプレイに送られる。第二の主要な構成部分は、MPEGアプリケーション/アプレット（MPEG App）100であり、この構成部分は、それぞれ、120を通して、外部オーサリング・ユニット130およびユーザ入力に、本発明のオーサリングAPIおよび機能APIをインターフェイスする。さらに、ジャバ仮想機械、およびジャバ・メディア・フレームワーク（JVMおよびJMF）110は、BIFSおよびメディアデコーダ、バッファ、コンポジットおよびレンダリング装置170に接続するための下支え基盤として使用され、また、シーングラフAPI150（MPEGにより提供され、本発明で使用する）およびデコーダAPIを通して、BIFSおよびメディアデコーダ、バッファ、コンポジットおよびレンダリング装置170に直接インターフェイスする。

10

【 0 0 2 1 】

図2は、図1の種々のブロック、構成部分およびインターフェイスのさらに詳細な図面である。この図では、オーサリング・ユニット130が、ライン200上で、ライン205を通してインターフェイスするユーザ入力140から独立して、MPEG App 100にインターフェイスしている。この図は、また各インターフェイス、オーサリングAPI 290および機能API 295も含む。さらに、この図では、MPEG App 100およびその下のJVMおよびJMF 110が、ライン215を通してBIFSデコーダおよびシーングラフ225に働きかけ、207を通してシーングラフAPI 210にインターフェイスしている。BIFSデコーダおよびシーングラフ225は、ライン260、261、262を通して多数の例示としてのメディアデコーダ270、271、272を制御し、また（ライン268および269を通して）コンポジット282およびレンダリング装置284を制御する。MPEG App 100に関連するJVMおよびJMF 110は、同様に、各ライン263、264、265を通して、メディアデコーダ270、271および272を制御する。図2について、種々のプログラムを使用する制御装置およびインターフェイスを説明してきた。

20

30

【 0 0 2 2 】

図2の残りの部分は、MPEG-4のパラメトリックシステムを詳細に示す。この図の頂部の、本発明のプログラムを使用するシステムおよび方法の動作について説明する。復号されるMPEG-4システムのビットストリームは、チャンネル155を通して、ネットワーク/記憶装置配布インターフェイスDMIF 160に送られる。前記インターフェイスDMIF 160は、前記ビットストリームをライン230を通して、Demux 165に送る。パッケージから取り出され、分離されたビットストリームは、シーン叙述情報を含む複数の部分からなり、BIFSデコーダおよびシーングラフ225に送られる。前記ビットストリームは、また種々の各メディアデコーダ用の他の部分も含み、それぞれ、ライン240、245および250、復号バッファ251、252および253、ライン255、256および257を通して、メディアデコーダ270、271および272に送られ、前記メディアデコーダは、合成バッファ276、277および278への入力である復号したメディアを、ライン273、274および275に出力する。ライン279上のバッファ276の出力は、ライン280上のバッファ277の出力およびライン281上のバッファ278の出力と一緒に、コンポジットへ送られる。

40

【 0 0 2 3 】

50

図には、(復号バッファ251、252、253、デコーダ270、271、272および合成バッファ276、277、278を通しての)三組のメディア復号動作しか示していないが、実際には、メディアデコーダの数は、一つでもいいし、また幾つでも必要なだけ使用することができる。コンポジタ282は、BIFSシーングラフ(およびおそらくはユニット入力)に基づいて、復号したメディアを相互に配列し、シーンを合成する。前記情報は、ライン283を通して、レンダリング装置284へ送られる。レンダリング装置284は、ピクセルおよびオーディオ・サンプルを与え、それらをライン175を通して、ディスプレイ(図示していないスピーカを持つ)285に送る。

【0024】

図3は、特定の例として、ビジュアル復号を使用する本発明のメディア復号機能である。説明を簡単にするために、メディア復号を単に復号と呼ぶことにする。ベースAVオブジェクトまたはビデオ・デコーダ構造体の使用可能性に関していくつかの仮定が必要である。前記仮定は、デフォルトまたはプレースホルダ動作を含む前記抽出クラスが、その構造体を無視することによって、多くの場合、拡張される、オブジェクト指向プログラミングの状況としては普通のものである。

【0025】

復号API220は、インターフェイス、より詳しく説明すると、ビジュアル復号API301である。前記ビジュアル復号API1301を使用して、多数の異なるビジュアル・デコーダ320を接続することができる。前記接続は、ブロック選択復号ロジック(SDL)306を通して行われる制御であると見なすことができる。図では、前記接続は、ロジックの他のいくつかの部分と一緒に、BIFSデコーダ・ロジック305、BIFSデコーダおよびシーングラフ225の一部または構成部分に属する。BIFS復号ロジック305は、制御ライン307を通してベース・ビデオ・デコーダ313のような種々のビジュアル・デコーダを制御し、制御ライン308を通して時間強化デコーダ314を制御し、制御ライン309を通して空間強化デコーダ315を制御し、制御ライン310を通して出力区分デコーダ316を制御し、制御ライン311を通して画像肌理デコーダ317を制御し、ライン312を通してメッシュ幾何学/運動デコーダ318を制御する。

【0026】

復号されるビットストリームは、ライン319(図2のメディアデコーダ入力255または256または257に対応する)を通して、適当なデコーダに送られ、復号された出力は、ライン325(図2のメディアデコーダ出力273または274または275に対応する)上で入手することができる。ベース・ビデオ・デコーダ313は、スケール不可能なビデオを復号し、空間強化デコーダ315は、空間スケール可能性ビデオ層を復号し、時間強化デコーダ314は、時間スケール可能性ビデオ層を復号し、データ区分デコーダ316は、データ分割ビデオ層を復号し、画像肌理デコーダ317は、静止画像の肌理の空間/SNRスケール可能な層を復号し、メッシュ幾何学/運動デコーダ318は、オブジェクトの動きにより、波形メッシュ・ノード位置およびこれらノードの運動を復号する。

【0027】

前記デコーダは、当業者には周知のMPEG-4ビジュアル規格により指定される。柔軟で、一貫した方法でMPEG-4ビジュアル・デコーダに、アクセスするのに使用する本発明の前記分類を、以下に詳細に説明する。

【0028】

復号API

```
Class mpgj.dec.BaseAVObject
```

```
public class BaseAVObject
```

これは、ベースAVオブジェクトストリームの復号を可能にする、基本的なクラスである。

【0029】

構成装置

```

public BaseAVObject()
方法
public void startDec()
データ復号の開始
public void stopDec()
データ復号の停止
public void attachDecoder(Mp4Stream baseStrm)
復号を行う有効なMPEG-4ストリームを復号するための、準備中のベース・ストリームへのデコーダの取り付け。
【0030】
ビジュアル復号API
Class mpgj.dec.Mp4Decoder
public class Mp4VDecoder
はビデオ・デコーダに拡張する。
このクラスは、ビデオ・デコーダ、抽出クラス(図示せず)を拡張する。このクラスは、種々のタイプのビジュアル・ビットストリームを復号するための方法を含む。
【0031】
構成装置
public Mp4VDecoder()
方法
public VObject baseDecode(Mp4Stream baseStrm)
MPEG-4ビデオストリーム、baseStrmを復号し、復号したビジュアルオブジェクト、VObjectを返す。
public VObject sptEnhDecode(Mp4Stream enhStrm)
空間強化MPEG-4ビデオストリーム、enhStrmを復号し、復号したビジュアルオブジェクト、VObjectを返す。
public VObject tmpEnhDecode(Mp4Stream enhStrm)
時間強化MPEG-4ビデオストリーム、enhStrmを復号し、復号したビジュアルオブジェクト、VObjectを返す。
public VObject snrEnhDecode(Mp4Stream enhStrm, int level)
レベルに従って、snr強化MPEG-4ビデオストリーム、enhStrmを復号し、復号したビジュアルオブジェクト、VObjectを返す。
public VObject dataPartDecode(Mp4Stream enhStrm, int level)
レベルに従って、データ分割MPEG-4ビデオストリーム、enhStrmを復号し、復号したビジュアルオブジェクト、VObjectを返す。
public VObject trickDecode(Mp4Stream trkStrm, int mode)
モードに従って、トリックストリームをスキップおよび復号し、復号したビジュアルオブジェクト、VObjectを返す。
public MeshObject meshAuxDecode(Mp4Stream auxStrm)
MPEG-4補助ビデオストリームを復号し、メッシュ幾何学および動作ベクトルを含む、メッシュオブジェクト、MeshObjectを返す。
【0032】
図4は、それに対して、インターフェイスが他の分類のAPIにより定義される、多数の例示としての機能を使用する、本発明のいくつかの特徴の機能である。前記機能API

```

10

20

30

40

50

295は、いくつかのインターフェイスを示す。より詳しく説明すると、トリックモード機能(401)用のインターフェイス、方向性機能用のインターフェイス(402)、透明機能用のインターフェイス(403)、ホットオブジェクト機能用のインターフェイス(404)、および順送り機能用のインターフェイス(405)を示す。各APIを使用して、多数の異なるデコーダを接続することができる。一例として、再びビジュアルデコーダを使用する。前記接続は、この図の場合、BIFSデコーダおよびシーングラフ225および/またはMPEG-4 App100の構成部分である、APP/BIFS Dec(oder)に属する、いくつかのブロック選択復号ロジック(SDL)416、417、418、419、420を通しての制御と見なすことができる。APP/BIFS復号ロジック415は、制御ライン421、422、424、425を通して、ベース・ビデオ・デコーダ313のような種々のビジュアル・デコーダを制御し、制御ライン423および426を通して、時間強化デコーダ315を制御し、制御ライン427を通して、空間強化デコーダ315を制御し、制御ライン429を通して、データ分割デコーダ316を制御し、制御ライン430を通して、画像肌理デコーダ317を制御し、ライン428を通して、メッシュ幾何学/運動デコーダ318を制御する。

【0033】

復号するビットストリームは、ライン431(図2のメディアデコーダ入力255または256または257に対応する)を通して、適当なデコーダへ送られ、復号した出力は、ライン445(図2のメディアデコーダ出力273または274または275に対応する)上で入手することができる。多くの場合、ビジュアルオブジェクトに関するユーザ機能は、一つまたはそれ以上のビジュアル・デコーダを使用することにより、実現することができることを理解することが重要である。前記SLDは、各ビジュアルオブジェクトを復号するために接続する特定のデコーダを選択するためばかりでなく、一つのビットストリームのために使用したデコーダ、および使用されるまたは使用されない特定の時間を選択するためにも使用される。図には多数のSDL416、417、418、419、420を示したが、各SLDは各機能に対応する。この図の各SDLは一つの制御入力を持つが、しかし、それはいくつかの可能性のある制御出力の中の一つである。もっと分かりやすく説明すると、図3の場合には、ベース・ビデオ・デコーダ313は、スケールすることができないビデオを復号し、空間強化デコーダ314は、空間スケール可能性ビデオ層を復号し、時間強化デコーダ315は、時間スケール可能性ビデオ層を復号し、データ分割デコーダ316は、データ分割ビデオ層を復号し、画像肌理デコーダ317は、静止画像肌理の空間/SNRスケール可能層を復号し、メッシュ幾何学/運動デコーダ318は、オブジェクトの運動により、ワイヤフレーム・メッシュ・ノードおよび前記ノードの運動を復号する。前記デコーダは、ここでもMPEG-4ビジュアル規格により指定される。柔軟でしっかりした方法で前記機能を達成するために使用する、本発明のAPIのこの分類についての詳細について以下に説明する。

【0034】

機能API

下記のAPIは、種々のユーザ相互作用機能のためのものである。

順送りAPI

```
Class mpgj.func.ProgAVObject
public class ProgAVObject
BaseAVObjectの拡張。
```

ProgAVObjectにより、ユーザ制御下のAVオブジェクトの品質を順送りに洗練することができる。現在、ビジュアルオブジェクトは、静的なもの(任意に形成したオブジェクトの時間どおりの一例である、静止画像「vops」、ビデオオブジェクト面と見なされる。前記形が長方形である場合には、vopは一つのフレームと同じものである。)

【0035】

構成装置

```

public ProgAVObject()
方法
public void startDec()
データ復号の開始
public void stopDec()
データ復号の停止
public void pauseDec()
データ復号の一時的な中止
public void resumeDec()
現在の一時中止の状態からの、データ復号の再開 10
public int selectProgLevel()
変換(DCTまたはウェレット)係数の復号が行われるまでのレベルの選択。一つのレ
ベルは、走査順序中のある位置までの係数である。
public void attachDecoder(Mp4Stream srcStr
rm int progLvl)
有効なMPEG-4ストリームを復号するための、準備中のsrcStrmへのデコーダ
の取り付け、および復号が行われるまでのプログラム・レベルの指定。
public void offsetStream(Mp4Stream srcStr
m, ulong offset)
srcStrmへの、復号開始目標としてのオフセットの挿入可能。実際には、現実の目標
位置は、必要な目標より遠い場合があり、ストリーム内の有効なエントリの位置により異
なる。 20
【0036】
Hot Object/Region API
このAPIにより、ホット(能動)AVオブジェクトと相互作用を行うことができる。
前記APIは、一つのオブジェクトの能動領域と、相互作用を行うことができるように拡
張することができる。このAPIは、AVオブジェクトの空間解像度強化、品質強化、時
間的品質強化のような、一つまたはそれ以上の高度な機能を可能にするためのものである
。実現する実際の強化は、(マウス・クリック/メニューを通しての)ローカルおよび遠
隔地からのユーザ相互作用および強化ストリーム、および使用することができる強化デコ
ーダにより異なる。 30
【0037】
Class mpgj.func.HotAVObject
public class HotAVObject
BaseAVObjectの拡張
HotAVObjectは、そのオブジェクトがホットオブジェクトである場合に、A
VObjectの強化動作をトリガするクラスである。それ故、ホットオブジェクトは、
必要な場合にトリガされる、それ自身に関連するいくつかの強化ストリームを持つ。この
クラスは、ベース(層)ストリームを復号するために主として使用される、BaseAV
Objectを拡張する。さらに、ホットオブジェクトの定義を、問題の領域(Key R
egion)を含むように拡張することができる。 40
【0038】
構成装置
public HotAVObject()
方法
public void startDec()
データ復号の開始
public void stopDec()
データ復号の停止
public void pauseDec() 50

```

データ復号の一時的な中止

```
public void resumeDec()
```

現在の一時中止の状態からの、データ復号の再開

```
public int selectHotType()
```

(空間、品質、時間等の)強化のタイプの選択

```
public Mp4Stream enhanceObject(int type)
```

必要な強化ストリームを入手するための、選択した強化タイプの使用

```
public void attachDecoder(Mp4Stream srcStrm  
int type)
```

有効なMPEG-4ストリームを復号するための、準備中のsrcStrmへのデコーダ
の取り付け、および復号が行われるタイプの指定

```
public void offsetStream(Mp4Stream srcStrm,  
ulong offset)
```

srcStrmへの、復号開始目標としてのオフセットの挿入可能。実際には、現実の目標
位置は、必要な目標より遠い場合があり、ストリーム内の有効なエントリの位置により異
なる。

【0039】

方向性API

このAPIにより、方向を感知するAVオブジェクトとの相互作用を行うことができる
。前記APIは、静的ビジュアルオブジェクト(静止vop)、動的ビジュアルオブジェ
クト(運動するvop)および方向性を持つ音声およびオーディオをサポートする。ビ
ジュアルオブジェクトの場合には、このAPIは、視点の選択を可能にし、対応するビット
ストリームだけが復号され、復号されたデータは、コンポジットへ送られる。音声オブ
ジェクトの場合には、必要な音声点により、アナログ動作が行われる。現在のところ、予め定
めた方向性を持つ選択が行われる。

【0040】

```
Class mpgj.func DirecAVObject
```

```
public class DirecAVObject
```

BaseAVObjectの拡張。

DirecAVObjectは、(予め定めた量子化した方向の形での)空間の、x -
y - z位置に回答するオブジェクトの生成を可能にする。ユーザ相互作用により、一つま
たはそれ以上の視点に対応するvopが、必要に応じて復号されるように、ビットスト
リームが、AVオブジェクトとして符号化された多数の静的ビジュアルvopからできてい
ると仮定すると、もっと容易に説明することができる。前記クラスは、動的AVObje
ctを符号化するのにも適している。

【0041】

構成装置

```
public DirecAVObject()
```

方法

```
public void startDec()
```

データ復号の開始

```
public void stopDec()
```

データ復号の停止

```
public void pauseDec()
```

データ復号の一時的な中止

```
public void resumeDec()
```

現在の一時中止の状態からの、データ復号の再開

```
public void loopDec()
```

この方法により、閉ループを形成する予め定めた順序の静的vopとしての、動的ビ
ジュアルオブジェクトのユーザ相互作用復号が可能になる。この方法はオーディオの場合にも

10

20

30

40

50

、適用することができる。ユーザの選択は、マウスのクリックまたはメニューにより行われる。

```
public int selectDirect()
```

方向（シーンの向き）の選択。オブジェクト上のホット点上またはメニュー上で、マウスをクリックすることにより、多数の予め定めた方向を使用することができ、選択を行うことができる。

```
public Mp4Stream enhanceObject(int orient)
```

必要な時間補助（強化）ストリームを入手するための、選択したシーンの向きの使用。

```
public void attachDecoder(Mp4Stream src stream int orient)
```

有効なMPEG-4ストリームを復号するための、準備中のsrc streamへの時間補助（強化）デコーダの取り付け、およびAVオブジェクトの選択したシーン方向の指定。

src streamへの、復号開始目標としてのオフセットの挿入可能。実際には、現実の目標位置は、必要な目標より遠い場合があり、ストリーム内の有効なエントリの位置により異なる。

【0042】

トリックモードAPI

トリックモードAPIは、強化されたトリックプレイ能力を使用することができるように、ユーザの制御の下での条件付き復号をサポートする。強化トリックプレイは、FFまたはFR、フリーズ・フレーム、ランダム・アクセスおよび逆方向再生のような、他の動作に対する異なる速度のような、VCR/CDプレーヤのような機能を使用できるようにするものであると見なすことができる。しかし、合成された全シーンに基づく他に、個々のAVオブジェクトに基づいて、MPEG-4がこれら能力を使用することができるという違いを持つ。

【0043】

```
Class mpgj.func.TrickAVObject
```

```
public class TrickAVObject
```

BaseAVObjectの拡張。

TrickAVObjectは、トリックプレイに適する復号を行うことができる、オブジェクトを形成するのに使用することができるクラスである。

【0044】

構成装置

```
public TrickAVObject()
```

方法

```
public void startDec()
```

データ復号の開始

```
public void stopDec()
```

データ復号の停止

```
public void pauseDec()
```

データ復号の一時的な中止

```
public void resumeDec()
```

現在の一時中止の状態からの、データ復号の再開

```
public void loopDec()
```

このクラスにより、種々の速度での順方向および逆方向への、src streamの選択した部分のユーザ相互作用復号が可能になる。

```
public boolean selectDirec()
```

復号方向の選択。トリック復号が（正規の）順方向に行われる場合には、真に戻る。逆方向のトリック復号が選択された場合には、偽に戻る。

```
public Mp4Stream enhanceObject(boolean decode direc)
```

10

20

30

40

50

`decDirec`により指定した方向へ復号するMPEG-4ストリームの入手。
`public void attachDecoder(Mp4Stream srcStream, int decDirec)`

有効なトリックモードのMPEG-4ストリームを復号するための、準備中の`srcStream`にトリックデコーダの取り付け、復号方向の指定。

`public void offsetStream(Mp4Stream srcStream, ulong offset)`

`srcStream`への、復号開始目標としてのオフセットの挿入可能。実際には、現実の目標位置は、必要な目標より遠い場合があり、ストリーム内の有効なエントリの位置により異なる。

10

【0045】

透明API

透明APIは、ユーザ制御下の一つのオブジェクトの、複数の領域の選択的な復号をサポートする。ビジュアルオブジェクトの場合には、そのオブジェクト内の他のピクセルの透明度を変化させることによって、大きなオブジェクトをいくつかのより小さな領域に分割するという方法で、符号化が行われると仮定する。問題の領域に属さないピクセルは、符号化されている領域内に存在しない選択したキー・カラーを、前記ピクセルに割り当てることにより符号化される。このAPIにより、いくつかのまたは全部の領域を符号化することができるように、ユーザの制御下で符号化を行うことができる。さらに、問題の領域の場合には、空間および時間的品質を改善するために、強化ビットストリームが必要になる場合がある。各領域に対する前記キー・カラーは、コンポジットに対して識別される。ユーザは、帯域幅/計算資源が限定されていたり、オブジェクトのいくつかの部分が隠れていたりして必要がないため、または他の領域ないに画像を削除するかその品質を犠牲にして、特定の領域に必要な遙かに高い品質のため、ユーザが、全部の領域の復号を必要としない場合がある。キー・カラーを使用する前記プロセスは、放送分野の「クロマキー」技術に類似している。

20

【0046】

`Class mpGj.svs.TranspAVObject`

`public class TrasnspAVObject`

`BaseAVObject`の拡張

30

`TrasnspAVObject`は、透明情報によりオブジェクトを形成するために使用することができるクラスである。音声およびビジュアルオブジェクトタイプの両方が処理される。

【0047】

構成装置

`public TranspAVObject()`

方法

`public void startDec()`

データ復号の開始

`public void stopDec()`

データ復号の停止

40

`public void pauseDec()`

データ復号の一時的な中止

`public void resumeDec()`

現在の一時中止の状態からの、データ復号の再開

`public int getRegion()`

リストの形のメニュー内の番号、または(同様に番号に翻訳される)ホット・ポイントをクリックすることによる領域の選択。

`public Mp4Stream enhanceObject(int type, int regnum)`

50

region regnumに対する必要な強化ストリームを入手するための、選択した強化タイプの使用。

```
public void attachDecoder(Mp4Stream srcStrm, int type, int regnum)
```

一つの領域およびそのキー・cカラーを復号するための準備中のsrcStrmにデコーダの取り付け。

```
public void offsetStream(Mp4Stream srcStrm, ulong offset)
```

srcStrmへの、復号開始目標としてのオフセットの挿入可能。実際には、現実の目標位置は、必要な目標より遠い場合があり、ストリーム内の有効なエントリの位置により異なる。

10

【0048】

図5は、それに対して、インターフェイスが他の分類のAPIで定義される、ストリーム編集の一例を使用する本発明のオーサリング機能である。オーサリングAPI290は、オーサリング関連のインターフェイス、より詳しく説明すると、ストリーム編集API(501)を表す。前記APIを使用することにより、MPEGApp(100)またはBIFSデコーダおよびシーングラフ(225)が使用するための、ビットストリームを編集/修正することができる。API501は、制御ライン505を通して、MPEGAPP100を制御し、制御ライン506を通してBIFSデコーダおよびシーングラフ225を制御する。それ故、ストリーム編集APIは、種々のオーディオ・ビジュアルオブジェクトおよびBIFSシーン叙述を含む、MPEG-4の編集/修正を助けることができる。ストリーム編集APIの他に、他のAPIを使用することができるオーサリングも行うことができるが、本発明は指定しない。ストリーム編集APIの詳細を下記に説明する。

20

【0049】

オーサリング

下記のAPIは、MPEG-4ビットストリームを部分的にオーサリングするためのものである。

【0050】

ストリーム編集API

```
Class mpgj.util.StreamEdit public class StreamEdit
```

30

このクラスは、MPEG-4ストリームの内容および修正を決定することができる。アクセス、コピー、追加、置き換え、削除のような動作および他の動作がサポートされる。

【0051】

構成装置

```
public StreamEdit()
```

方法

```
public int[] getObjectList(Mp4Stream rcStrm)
```

40

ストリーム内のオブジェクトのリストの返還。返還されたオブジェクトは、ビットストリーム内のオブジェクトの累積表となる。

```
public boolean replaceObject(Mp4Stream srcStrm, ulong srcObjid, Mp4Stream destStrm, ulong destObjid)
```

ストリーム内のオブジェクトid、destObjidを持つオブジェクトの発生、srcStrm内のオブジェクトid、srcObjidを持つオブジェクトの対応する発生による置き換え。オブジェクト表はそれに従って更新される。更新が成功した場合には、動作は真に戻り、更新が失敗した場合には偽に戻る。

```
public boolean replaceObjectAt(Mp4Stream
```

50

`srcstrm`、`ulongsrcobjid`、`ulongm`、`Mp4Streamdeststrm`、`ulongdestobjid`、`ulongn`)

置き換えを開始する位置を除いて、`replaceObject()`と同じ意味論が指定される。`destobjid`の n 番目の発生からの宛先オブジェクトの、`srcobjid`の m 番目の発生からのソースオブジェクトによる置き換え。 $m = n = 0$ の場合には、`replaceObject()`と同じことが行われる。

`public boolean containObjectType(Mp4Stream srcstrm, ulong objtype)`

`srcstrm`が`objtype`のオブジェクトを含んでいる場合には、真に戻り、そうでない場合には偽に戻る。

10

`public boolean addObjects(Mp4Stream srcstrm, ulong srcobjid, Mp4Stream deststrm)`

`deststrm`への`srcstrm`からの`srcobjid`の追加。成功の場合は真に戻り、そうでない場合には偽に戻る。

`public boolean addObjectAt(Mp4Stream srcstrm, ulong srcobjid, Mp4Stream deststrm, ulong destobjid, ulong n)`

`destobjid`の n 番目の発生の後から始まる`deststrm`への、`srcstrm`からの`srcobjid`の追加。成功の場合は真に戻り、そうでない場合には偽に戻る。

20

`public boolean copyObjects(Mp4Stream srcstrm, ulong srcobjid, Mp4Stream deststream, destobjid)`

`srcstrm`内の`srcobjid`を持つオブジェクトを、新しいオブジェクト`id`、`destobjid`を持つ`deststream`へコピー。`deststrm`が存在しない場合には、生成される。存在する場合には重ね書きされる。この動作は、以降の動作に対する多重化されていないストリームから、基本的ストリームを生成するために使用される。成功の場合は真に戻り、そうでない場合には偽に戻る。

`public boolean deleteObjects(Mp4Stream deststrm, ulong destobjid)`

30

`deststrm`内の`destobjid`を持つストリームオブジェクトの削除。また、すべての合成情報の除去。成功の場合は真に戻り、そうでない場合には偽に戻る。

`public boolean spliceAt(Mp4Stream deststrm, ulong destobjid, ulong n, Mp4Stream srcstrm)`

`destobjid`の n 番目の発生後の`deststrm`のスライス、および`srcstrm`のペースト。成功の場合は真に戻り、そうでない場合には偽に戻る。

【0052】

まとめると、図1 - 図5のAPIに反映された、一組のライブラリ機能を含む柔軟なシステムおよび方法は、符号化されたメディア・ストリームをリモート・ラップトップ・パソコンまたは他の装置のような復号ターミナル資源とマッチさせることができる、新しいレベルの適応性を提供する。さらに、本発明は、また適当なデコーダおよび符号化されたメディアオブジェクトビットストリームの選択的な復号と一緒に、高度の新しい機能を可能にする、ユーザ相互作用に対するサポートも含む。

40

【0053】

本発明を実行する場合には、定義のAV関連機能を含む分類が導入され、すべての汎用で一貫している方法で、埋設されたオーディオ・ビジュアルオブジェクトの復号と合成との間の、簡単でありながら、より複雑な相互作用を行うことができるように、一組のAPIが設定される。

【0054】

50

本発明のシステムおよび方法の前記説明は、単に例示としてのものに過ぎず、当業者ならその構造および実行を種々に変更することができるだろう。例えば、三つの分類の入力、出力およびマッピング機能のコンパクトで汎用装置を説明してきたが、ネットワーク、アプリケーションまたは他のニーズの変更により、APIにいくつかの機能を追加したり、取り除いたりすることができる。本発明の範囲は、下記の請求の範囲だけにより制限される。

【図面の簡単な説明】

【0055】

【図1】本発明の一実施形態を示すシステムのハイレベルのブロック図である。

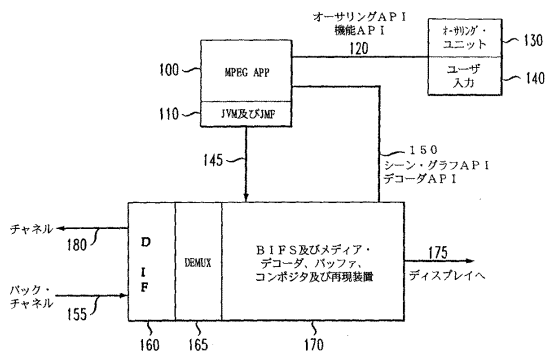
【図2】本発明の前記実施形態を詳細に示す前記システムのブロック図である。

【図3】本発明のビジュアル復号のためのインターフェイス方法である。

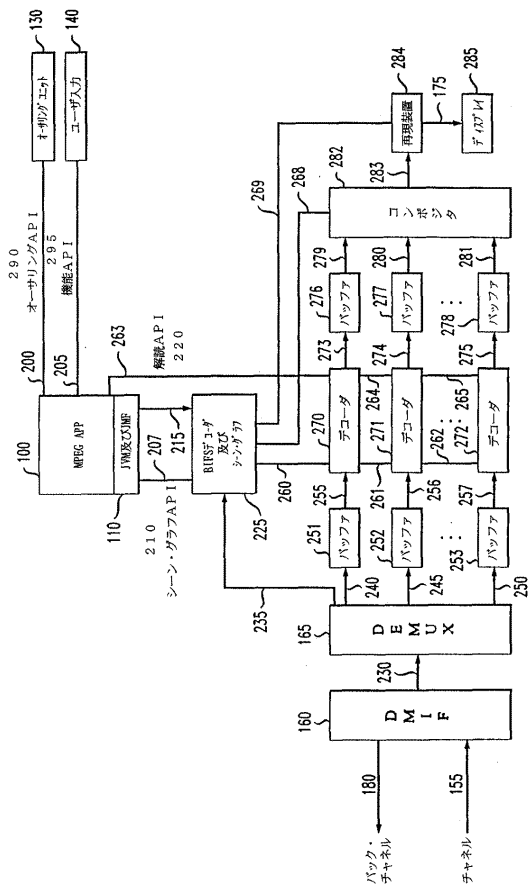
【図4】本発明の機能用のインターフェイス方法である。

【図5】本発明のオーサリングのためのインターフェイス方法である。

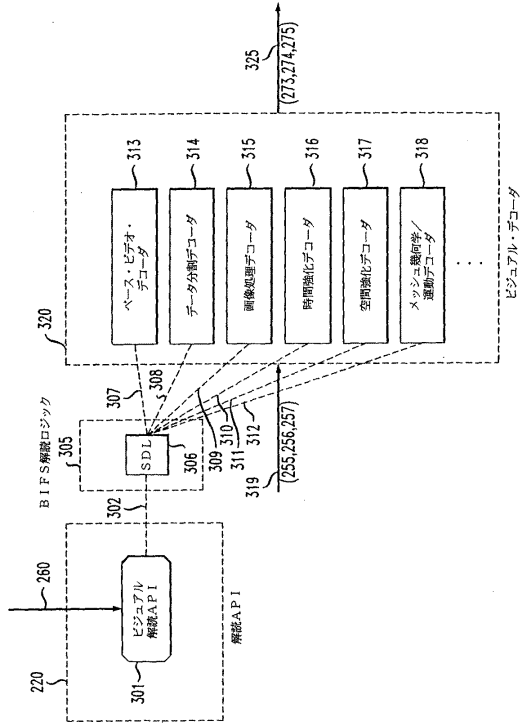
【図1】



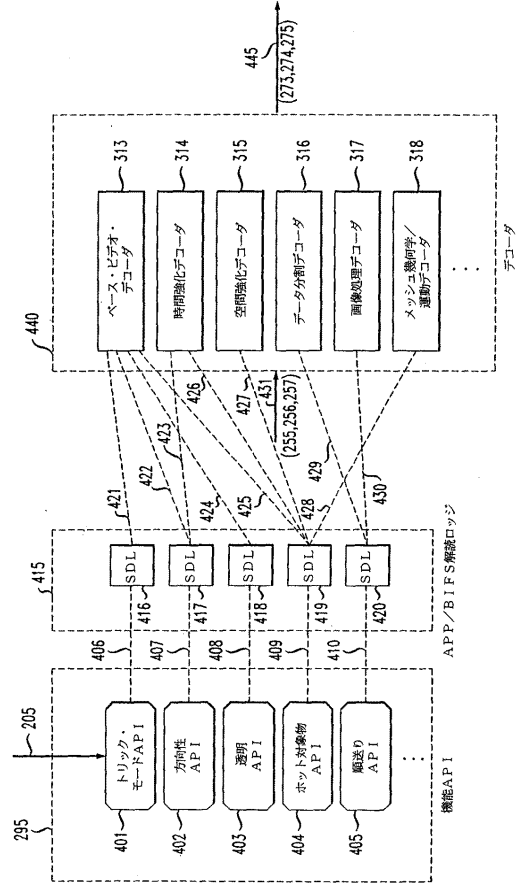
【図2】



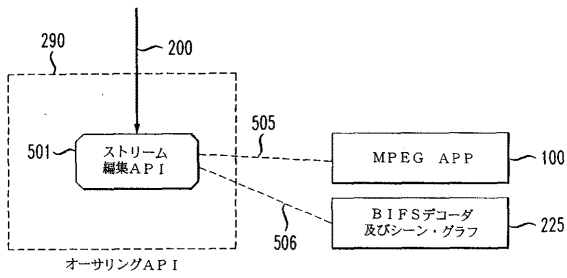
【図3】



【図4】



【図5】



フロントページの続き

(73)特許権者 507352341

ザ トラストィーズ オブ コロンビア ユニヴァーシティ イン ザ シティ オブ ニューヨーク

アメリカ合衆国 10027 ニューヨーク, ニューヨーク, ウェスト ワンハンドレッド シックスティーンズ ストリート 535, ロウ メモリアル ライブラリー 110

(74)代理人 100094112

弁理士 岡部 譲

(74)代理人 100064447

弁理士 岡部 正夫

(74)代理人 100085176

弁理士 加藤 伸晃

(74)代理人 100104352

弁理士 朝日 伸光

(72)発明者 エレフゼリアディス, アレキサンドロス

アメリカ合衆国 10027 ニューヨーク, ニューヨーク, アパートメント 4エル, リヴァーサイド ドライヴ 560

(72)発明者 ファング, イーハン

アメリカ合衆国 10025 ニューヨーク, ニューヨーク, アパートメント 2イー, ウェストワンハンドレッド ツェルヴス ストリート 539

(72)発明者 カルヴァ, ハリ

アメリカ合衆国 10027 ニューヨーク, ニューヨーク, アパートメント 8イー, ウェストワンハンドレッド ナインティーンズ ストリート 419

(72)発明者 プリ, アチュル

アメリカ合衆国 10463 ニューヨーク, リヴァーデール, ナンバー 1イー, ワールド アヴェニュー 3660

(72)発明者 シュミット, ロバート エル.

アメリカ合衆国 07731 ニュージャージー, ホーウェル, オーク グレン ロード 333

合議体

審判長 乾 雅浩

審判官 梅本 達雄

審判官 渡邊 聡

- (56)参考文献 Laier, J., et al, Content-Based Multimedia Data Access in Internet Video Communication, Proc. of 1st Workshop on Wireless Image/Video Communications, 1996年 9月, P.126-133
A. Puri, R. L. Schmidt, A. Eleftheriadis, H. Kallva, Y. Fang, APIs for MPEG-4 Systems, ISO/IEC JTC1/SC29/WG11MPEG97/2133, 1997年 4月
Fang, Y., Eleftheriadis, A., Puri, A., and Schmidt, R., Adding Bitstream Input/Output Library in Java, ISO/IEC JTC1/SC29/WG11MPEG97/M2063, 1997年 4月 1日