

(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2010-226278
(P2010-226278A)

(43) 公開日 平成22年10月7日(2010.10.7)

(51) Int.Cl.	F 1	テーマコード (参考)
H04N 5/91 (2006.01)	H04N 5/91	5 C052
H04N 5/76 (2006.01)	H04N 5/76	5 C053
H04N 5/93 (2006.01)	H04N 5/93	5 D044
G11B 20/10 (2006.01)	H04N 5/93	5 D110
G11B 27/00 (2006.01)	G11B 20/10	311

審査請求 未請求 請求項の数 7 O L (全 19 頁) 最終頁に続く

(21) 出願番号	特願2009-69470 (P2009-69470)	(71) 出願人	000005821 パナソニック株式会社 大阪府門真市大字門真1006番地
(22) 出願日	平成21年3月23日 (2009.3.23)	(74) 代理人	100097445 弁理士 岩橋 文雄
		(74) 代理人	100109667 弁理士 内藤 浩樹
		(74) 代理人	100109151 弁理士 永野 大介
		(72) 発明者	河口 宜史 大阪府門真市大字門真1006番地 パナソニック株式会社内
		(72) 発明者	富永 喜代司 大阪府門真市大字門真1006番地 パナソニック株式会社内

最終頁に続く

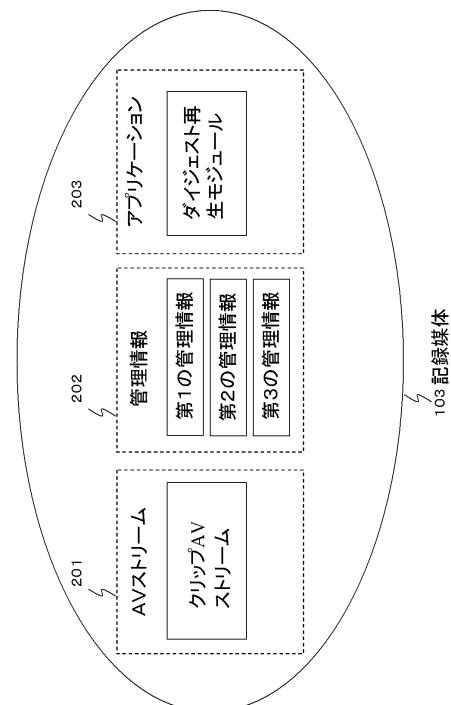
(54) 【発明の名称】記録装置、方法、プログラム、及び媒体

(57) 【要約】

【課題】映像の記録装置で記録した記録媒体を再生装置で再生する際のユーザの利便性を向上させる。

【解決手段】映像の記録装置は、映像を撮影して記録する際に、映像データであるAVストリーム201と、AVストリーム201の属性情報を示す第1の管理情報と、撮影開始、停止により生成された複数のAVストリームの再生順序を示す第2の管理情報と、AVストリーム201の特定区間に対する映像の有効性を示す第3の管理情報から構成される管理情報202と、管理情報202の情報を基にAVストリーム201中の重要区間、無効区間の判定をインタラクティブに行うことが可能なプログラムで、無効シーンを省略した再生、任意の再生時間に重要区間を組み合わせたダイジェスト再生をすることが可能なダイジェスト再生モジュールを含むアプリケーション203を記録する。

【選択図】図2



【特許請求の範囲】**【請求項 1】**

映像を撮影して記録媒体に記録する記録装置において、
撮影開始および撮影停止の指示を入力する指示入力部と、
記録装置の撮影状態を信号に変換する信号変換部と、

前記指示入力部および前記信号変換部の情報から映像の重要度を解析する解析部と、
ビデオデータおよびオーディオデータを多重化してAVストリームを生成する多重化部
と、

前記多重化部で生成されたAVストリームの属性情報を示す第1の管理情報と、前記指
示入力部で入力された指示に係わる撮影開始、停止により生成された複数のAVストリーム
の再生順序を示す第2の管理情報と、前記解析部で解析された情報とを基に該AVストリーム
の特定区間にに対する映像の重要度を示す第3の管理情報とを生成する管理情報生成部と、
10

前記多重化部で生成されたAVストリームと前記管理情報生成部で生成された管理情報
の内容を基に、該AVストリームについて所定の再生条件によりダイジェスト再生するこ
とが可能なアプリケーションと、該AVストリームと、該管理情報とを記録媒体に記録す
る記録部とを

備えることを特徴とする記録装置。

【請求項 2】

前記管理情報生成部は、前記指示入力部による撮影開始および撮影停止操作、撮影中のカ
メラ制御情報と、前記信号変換部による手ぶれ・地面撮り判定または、人物と人物の有無
の識別、または、歓声の大きさまたはトーンの変化、のいずれかに関する情報を第3の管
理情報として生成することを特徴とする、請求項1に記載の記録装置。
20

【請求項 3】

該所定の再生条件とは、再生時間の指定、もしくは、ハイライト区間のみの指定、もしくは、
手ぶれ、地面撮りシーンなどの無効区間を省いた再生区間の指定、もしくは、指定した人物
または人物の存在する再生区間の指定、あるいは、これらを組み合わせた再生条件と
することを特徴とする、請求項1に記載の記録装置。

【請求項 4】

該アプリケーションとは、複数の再生環境に対応するために、複数の実行モジュール、あるい
は仮想マシン向けプログラミング言語で記述されたプログラムからなることを特徴と
する、請求項1に記載の記録装置。
30

【請求項 5】

映像を撮影して記録媒体に記録する記録方法において、
撮影開始および撮影停止の指示を入力する指示入力ステップと、
記録装置の撮影状態を信号に変換する信号変換ステップと、
前記指示入力ステップおよび前記信号変換ステップの情報から映像の重要度を解析する解
析ステップと、
ビデオデータおよびオーディオデータを多重化してAVストリームを生成する多重化ステ
ップと、
40

前記多重化ステップで生成されたAVストリームの属性情報を示す第1の管理情報と、前記指
示入力ステップで入力された指示に係わる撮影開始、停止により生成された複数のAVストリーム
の再生順序を示す第2の管理情報と、前記解析ステップで解析された情報とを基に該AVストリーム
の特定区間にに対する映像の重要度を示す第3の管理情報とを生成する管理情報生成ステップと、

前記多重化ステップで生成されたAVストリームと前記管理情報生成ステップで生成され
た管理情報の内容を基に、該AVストリームについて所定の再生条件によりダイジェスト
再生することが可能なアプリケーションと、該AVストリームと、該管理情報とを記録媒
体に記録する記録ステップと

を有することを特徴とする記録方法。
50

【請求項 6】

請求項 5 に記載の記録方法を含むプログラム。

【請求項 7】

請求項 6 に記載のプログラムを記録した媒体。

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、映像を撮影して記録媒体に記録する記録装置において、ユーザの利便性を向上させた視聴を可能とする記録装置、記録方法および記録プログラムに関する。

10

【背景技術】**【0002】**

従来のビデオカメラの場合、撮影済みの動画の再生方法としては、基本的に記録媒体（テープ、ディスクなど）に記録されている動画を、その撮影順に再生する方法しかない。

また、ユーザが撮影済みの動画について所定の時間で視聴するためには、撮影済みの動画の全貌を把握し、重要な部分を判断し、目的の時間に合わせるために数秒単位に映像を抽出、合成する編集作業が必要であり、非常に多くの手間と時間を要する作業である。

そこで、編集作業を効率的に行う発明として、特許文献 1 に記載されたものが知られている。動画の撮影時に収録したコンテンツ（映像、音声、データ）の映像に含まれる人物、物体を画像認識し、音声に含まれるキーワードなど音声の認識を行ない、それぞれの認識結果を文字データに変換してメタデータを生成する。そして、このメタデータを利用して、所望のシーンの映像を検索することにより編集作業を効率化するシステムである。

20

【特許文献 1】特開 2007 - 82088 号公報**【発明の開示】****【発明が解決しようとする課題】****【0003】**

このように、従来の技術では、映像の記録装置で記録された映像は、編集という膨大な過程を得なければ、ユーザが所望する時間の作品として視聴することができない。

本発明では、このような記録装置において、映像の記録後に編集作業をすることなくユーザの所望する条件（任意の時間、不要部分以外のシーン）で映像を再生することを可能とし、ユーザの利便性を向上させることが可能な記録装置、記録方法、記録プログラム、及び媒体を提供することにある。

30

【課題を解決するための手段】**【0004】**

上述した課題を解決するために、第 1 の発明は、映像を撮影して記録媒体に記録する記録装置において、撮影開始および撮影停止の指示を入力する指示入力部と、記録装置の状態を信号に変換する信号変換部と、指示入力部および信号変換部の情報から映像の重要度を解析する解析部と、ビデオデータおよびオーディオデータを多重化して AV ストリームを生成する多重化部と、AV ストリームの属性情報を示す第 1 の管理情報と、指示入力部で入力された指示に係わる撮影開始、停止により生成された複数の AV ストリームの再生順序を示す第 2 の管理情報と、解析部の情報を基に AV ストリームの特定区間にに対する映像の有効性を示す第 3 の管理情報とを生成する管理情報生成部と、AV ストリームと管理情報に加えて、管理情報の内容を基に、AV ストリームについて所定の再生条件によりダイジェスト再生することが可能なアプリケーションを記録媒体に記録する記録部を備えることを特徴とする記録装置である。

40

また、第 2 の発明は、映像を撮影して記録媒体に記録する記録方法において、撮影開始および撮影停止の指示を入力する指示入力のステップと、記録装置の状態を信号に変換する信号変換のステップと、指示入力のステップおよび信号変換のステップの情報から映像の重要度を解析する解析のステップと、ビデオデータおよびオーディオデータを多重化して AV ストリームを生成する多重化のステップと、AV ストリームの属性情報を示す第 1 の管理情報と、指示入力ステップで入力された指示に係わる撮影開始、停止により生成され

50

た複数のAVストリームの再生順序を示す第2の管理情報と、解析のステップの情報を基にAVストリームの特定区間にに対する映像の重要度を示す第3の管理情報を生成する管理情報生成のステップと、AVストリームと管理情報に加えて、管理情報の内容を基に、AVストリームについて所定の再生条件によりダイジェスト再生することが可能なアプリケーションを記録媒体に記録する記録のステップを備えることを特徴とする記録方法である。

また、第3の発明は、AVストリームと、管理情報と、アプリケーションとが記録された記録媒体であって、AVストリームは、映像、音声またはデータのいずれかを含むコンテンツであって、管理情報は、AVストリームを再生制御するための情報であり、また、映像、音声または記録装置の状態をパラメータ格納した情報であり、記録装置は、撮影時に事前に決めてある規則により特徴ある区間を検出し、アプリケーションが解釈することが可能なデータ形式で管理情報に記録し、アプリケーションは、仮想マシン向けプログラミング言語で記述されたプログラムであり、管理情報を基に特徴のある区間をダイジェスト化した再生経路を動的に決定してAVストリームを再生することを特徴としている。アプリケーションは、任意の時間を指定することで、管理情報に記載されたパラメータから重要な区間を抽出して、指定した時間で再生する再生手段をさらに備えることが望ましい。アプリケーションは、管理情報に記載されたパラメータから撮影に失敗したと思われる不要な区間を省いて再生する再生手段をさらに備えることが望ましい。

【発明の効果】

【0005】

上述したように、ビデオデータおよびオーディオデータを多重化したAVストリームと、AVストリームの属性情報を示す第1の管理情報と、再生方法を示す情報を含む第2の管理情報と、解析部の情報を基に上記AVストリームの特定区間にに対する映像の重要度を示す第3の管理情報とからなる管理情報に加えて、管理情報の内容を基にAVストリームについて所定の再生条件によりダイジェスト再生することが可能なアプリケーションを記録媒体に記録することで、ユーザは撮影後に編集をすることなくダイジェスト再生をすることが可能な記録媒体を生成することができる。

【発明を実施するための最良の形態】

【0006】

(実施の形態)

以下、添付された図面を参照して、本発明の望ましい実施形態について詳細に説明する。先ず始めに、本発明の使用行為についての形態を説明する。図1は、本発明に係る記録媒体の、使用行為についての形態を示す図である。図1において、ビデオカメラ101は、撮影対象102（風景や人やペットなどの動物、車、建造物などの物）を入力映像として、記録媒体103にAVストリームを記録する。AVストリームを記録した記録媒体103は、再生装置104、リモコン105、テレビ103の構成で視聴することができる。このうちリモコン105は、再生（Play）、停止（Stop）を受け付けるキー、メニュー操作時にあたってのフォーカス移動操作を受け付けるMoveUpキー、MoveDownキー、MoveRightキー、MoveLeftキー、メニュー表示操作を受け付けるPop-upキー、数値入力を受け付けるNumericキーが存在する。本発明において記録媒体103は、データ記録可能ならゆる媒体を意味し、例えば、光ディスク、フラッシュメモリ、磁気テープなど記録方式に関らず、あらゆる媒体を包括する意味をもつ。以下では、本発明の説明の便宜上、記録媒体を光ディスク（Optical disc）、特に、Blu-ray Disc（ブルーレイディスク：登録商標）として説明するが、本発明の技術思想は、他の記録媒体にも同一に適用可能であることは明らかである。以上が本発明に係る使用形態についての説明である。

【0007】

図2は、本発明の記録媒体に記録されたデータの種類を示す図である。図1に示すように、本発明の記録媒体103は、AVストリーム201、管理情報202、アプリケーション203を含む。AVストリーム201は、ビデオデータおよびオーディオデータを多

重化したビットストリームで、クリップAVストリームと称される。ここで、クリップAVストリームとは、連続同期再生、すなわち実時間再生が保証された映像のまとまりの単位で、撮影開始、停止により生成される連續したビットストリームの単位のことをいう。一般的な撮影では、撮影開始、停止を繰り返すため、複数のクリップAVストリームが生成される。クリップAVストリームは、所定のファイルシステムによりファイルとして記録媒体に記録される。このファイルを、クリップAVストリームファイル（またはAVストリームファイル）と称する。管理情報202は、クリップAVストリームの属性情報を示す第1の管理情報と、撮影開始、停止により生成された複数のクリップAVストリームの再生順序を示す第2の管理情報と、クリップAVストリームの特定区間にに対する映像の有効性を示す第3の管理情報から構成される。

アプリケーション203は、管理情報202の情報を基にAVストリーム201中の重要な区間、無効区間の判定をインタラクティブに行うことが可能なプログラムで、無効シーンを省略した再生や、任意の再生時間に重要区間を組み合わせたダイジェスト再生をすることが可能なダイジェスト再生モジュールを含む。かかるアプリケーション203は、CまたはJava（登録商標）のようなプログラム言語で作成され、本発明では、特にxletのようなJava（登録商標）アプリケーションを中心に説明する。Java（登録商標）アプリケーションは、仮想マシンのヒープ領域（ワーカメモリとも呼ばれる）にロードされた1つ以上のxletプログラムからなる。このワーカメモリにロードされたxletプログラム、及び、データからアプリケーションは構成されることになる。

【0008】

図3は、図2に示した記録媒体103に記録されたデータの構造を説明するための図である。図2に示すように、本発明による記録媒体103に記録されたマルチメディアデータは、4層のレイヤよりなる。最も最下層のレイヤは、クリップAVストリーム301が配置されるレイヤである（便宜上、クリップレイヤと呼ぶ）。その上のレイヤはクリップAVストリーム301に対する再生箇所を指定するための、ムービープレイリスト303（Movie Play List）と、プレイアイテム304（Play Item）とが配置されるレイヤである（便宜上、プレイリストレイヤと呼ぶ）。さらにその上のレイヤは、ムービープレイリスト303（に対して再生順などを指定するコマンドからなるムービーオブジェクト305（Movie Object）及びインタラクティブな再生をするためのJava（登録商標）プログラムであるBD-J（Blu-ray Disc Java（登録商標））オブジェクト306（BD-J Object）が配置されるレイヤである（便宜上、オブジェクトレイヤと呼ぶ）。最上層のレイヤは、記録媒体103に格納されるタイトルなどを管理するインデックスペース307が配置される（便宜上、インデックスレイヤと呼ぶ）。

【0009】

クリップレイヤについて説明する。クリップAVストリーム301は、ビデオデータやオーディオデータがMPEG2 TS（トランスポートストリーム）の形式などに多重化されたビットストリームである。クリップAVストリーム301に関する属性情報は、本発明の第1の管理情報であり、クリップ情報302（Clip Information）としてファイルに記録される。

【0010】

クリップAVストリーム301と、対応するクリップ情報302をひとまとまりのオブジェクトと見なし、クリップ（Clip）と称する。

【0011】

プレイリストレイヤについて説明する。ムービープレイリスト303は、本発明の第2の管理情報であり、再生するAVストリーム301の指定と、指定されたAVストリーム301の再生箇所を指定する再生開始点（IN点）と再生終了点（OUT点）の集まりとから構成される。この再生開始点と再生終了点の情報を一組としたものは、プレイアイテム304（Play Item）と称される。

ムービープレイリスト303は、プレイアイテム304の集合で構成される。プレイアイ

10

20

30

40

50

テム 304 は撮影開始から撮影停止までの区間に相当する。プレイリストエクステンション 310 はムービープレイリスト 303 の拡張情報を記述する領域で、本発明の第 3 の管理情報である AVストリームの特定区間にに対する映像の有効性を示す情報を格納されている。

【0012】

オブジェクトレイヤについて説明する。ムービーオブジェクト 305 は、ナビゲーション命令語プログラムからなり、かかるナビゲーション命令語は、プレイリストの再生を始めるか、またはムービーオブジェクト間の転換あるいはユーザの選好によってプレイリストの再生を管理する。BD-J オブジェクト 306 は、Java(登録商標) プログラムによるオブジェクトで、無効シーンを省略した再生や、任意の再生時間に重要区間を組み合わせたダイジェスト再生を実現するためのプログラムであるアプリケーション 203 が格納されている。

10

【0013】

インデックスレイヤについて説明する。インデックスレイヤは、インデックステーブル 307 からなる。インデックステーブル 307 は、複数のタイトルとメニューとを定義するためのトップレベルのテーブルであって、あらゆるタイトルとメニューの開始位置情報を含んでいるので、タイトル検索やメニュー コールのようなユーザ入力を通じて選択されたタイトルやメニューを再生できる。また、記録媒体が再生装置に挿入されたとき、自動的に実行される最初に再生されるタイトルやメニューの開始位置情報を含む。

20

図 4 は、本発明による記録媒体 103 に記録されたファイルのディレクトリ構造を説明するための図である。インデックステーブル 307 は、index.bdmv という名称のインデックステーブルファイルとして保存され、ナビゲーションデータを含むムービーオブジェクト 305 は、MovieObject.bdmv という名称のムービーオブジェクトファイルとして記録される。また、動画の再生単位であるプレイリスト 303 は、PLAYLIST というディレクトリ直下に拡張子 mpls が付与されたプレイリストファイル (00000.mpls, 00001.mpls, 00002.mpls, , ,) として記録され、クリップインフォメーション 302 は、CLIPINF ディレクトリ直下に拡張子 clpi が付与されたクリップインフォメーションファイル (01000.clpi, 02000.clpi, 03000.clpi, , ,) として記録され、クリップ AVストリーム 301 は、STREAM というディレクトリ直下に拡張子 m2ts が付与されたクリップ AVストリームファイル (01000.m2ts, 02000.m2ts, 03000.m2ts, , ,) として記録される。特に、クリップ AVストリームファイルと対応するクリップインフォメーションファイルは、同じ 5 衔の数字からなった名称を有し、拡張子のみが異なる。そして、BD-J オブジェクト 306 は、BD-JO ディレクトリ直下に拡張子 bdjo が付与された BD-J オブジェクトファイルとして記録され、実際のプログラムは JAR ディレクトリ以下に記録される。

30

次に、図 4 で示した各ファイルのうち、本発明に関わりの深いものについて、C 言語の記述法に基づき詳細に説明する。図 5 は、プレイリストファイルの構造を表すシンタクスを示す図である。フィールド TypeIndicator は、32 ビット (4 バイト) のデータ長を有し、このファイルがプレイリストファイルであることを示す。フィールド TypeIndicator2 は、32 ビット (4 バイト) のデータ長を有し、このプレイリストファイルのバージョンを示す。フィールド PlayListStartAddress は、32 ビットのデータ長を有し、このシンタクス中のブロックblkPlayList() の開始アドレスを示す。フィールド PlayListMarkStartAddress は、32 ビットのデータ長を有し、このシンタクス中のブロックblkPlayListMark() の開始アドレスを示す。フィールド ExtensionDataStartAddress は、32 ビットのデータ長を有し、このシンタクス中のブロックblkExtensionData() の開始アドレスを示す。フィールド ExtensionDataStartAddress は、ブロックblkExtensionData() の開始アドレスを、プレイリストファイルの最初のバイトからの相対バイト数を表し

40

50

た値である。相対バイト数は、”0”から開始される。このフィールド `ExtensionDataStartAddress` の値が ”0” であれば、このプレイリストファイル内に、ブロック `blkExtensionData()` が存在しないことを示す。160 ビットのデータ長を有する領域 `reserved` を介してブロック `blkAppInfoPlayList()` が配される。ブロック `blkAppInfoPlayList()` は、次のブロック `blkPlayList()` に記述されるプレイリストのタイプ、再生制限などの情報が記述される。ブロック `blkPlayList()` は、プレイリストが記述される。ブロック `blkPlayListMark()` は、チャプタジャンプなどでジャンプされるポイントが記述される。ブロック `blkExtensionData()` は、所定の拡張データを格納可能とするためのブロックである。

図 6 は、ブロック `blkPlayList()` の構造を表すシンタクスを示す図である。フィールド `Length` は、32 ビットのデータ長を有し、このフィールド `Length` の直後からブロック `blkPlayList()` の最後までのデータ長を示す。フィールド `Length` に続けて 16 ビットのデータ長を有する領域 `reserved` が配され、次にフィールド `NumberOfPlayItems` が配される。フィールド `NumberOfPlayItems` は、16 ビットのデータ長を有し、このブロック `blkPlayList()` に含まれるプレイアイテムの数を示す。フィールド `NumberOfSubPaths` は、このブロック `blkPlayList()` に含まれるサブパスの数を示す。次の `for` ループ文に従い、フィールド `NumberOfPlayItems` で示される数だけ、プレイアイテムが記述されるブロック `blkPlayItem()` が記述される。`for` ループ文に基づくカウント数がブロック `blkPlayItem()` の識別子 `PlayItem_id` となる。ブロック `blkSubPath()` は、本発明と関連性が薄いので、説明を省略する。

図 7 は、ブロック `blkPlayItem()` の構造を表すシンタクスを示す図である。フィールド `Length` は、16 ビットのデータ長を有し、このフィールド `Length` の直後からブロック `blkPlayItem()` の最後までのデータ長を示す。フィールド `ClipInformationFileName` は、40 ビット (5 バイト) のデータ長を有し、このブロック `blkPlayItem()` が参照するクリップインフォメーションファイルのファイル名が示される。このプレイアイテムにおいて、フィールド `ClipInformationFileName` で示されるファイル名のクリップインフォメーションファイルが読み出される。フィールド `ClipCodecidifier` は、32 ビット (4 バイト) のデータ長を有し、このブロック `blkPlayItem()` によるプレイアイテムにおいて用いられるクリップ AV ストリーム 301 のコーデック方式を示す。フィールド `ConnectionCondition`、フィールド `RefToSTCID`、ブロック `blkUOMaskTable()`、フィールド `PlayItemRandomAccessFlag`、フィールド `StillMode`、フィールド `StillTime` は、本発明と関連性が薄いので、説明を省略する。ブロック `blkNSTable()` は、このブロック `blkPlayItem()` によるプレイアイテムが管理しているクリップ AV ストリームの記録媒体上の記録位置の属性などを管理している。

図 8 は、ブロック `blkPlayItem()` の構造を表すシンタクスを示す図である。フィールド `Length` は、32 ビットのデータ長を有し、このフィールド `Length` の直後からブロック `blkPlayListMark()` の最後までのデータ長を示す。フィールド `NumberOfPlayListMarks` は、16 ビットのデータ長を有し、このブロック `blkPlayListMark()` に含まれるプレイリストマークの数を示す。次の `for` ループ文に従い、フィールド `NumberOfPlayListMarks` で示される数だけプレイリストマークの情報が記述される。`for` ループ文内において、8 ビットのデータ長を有する領域 `reserve` に続けてフィールド `MarkType` が配される。フィールド `MarkType` は、8 ビットのデータ長を有し、マークのタイプを示す。プレイリストマークには、エントリマーク (`EntryMark`) およ

10

20

30

40

50

びリンクポイント(Link Point)の2タイプが定義されており、このフィールド Mark Typeにより、何れのタイプであるかが示される。チャプタを定義するためには、エントリマークを用いる。リンクポイントは、本発明の特定区間にに対する映像の有効性を示す第3の管理情報が存在することを定義するために使用する。フィールド RefTo Play Item IDは、16ビットのデータ長を有し、マークが打たれるプレイアイテムを参照する識別情報 Play Item_idが記述される。フィールド Mark TimeStampは、32ビットのデータ長を有し、マークが打たれるポイントを示すタイムスタンプが記述される。フィールド Entry ESPIDは、16ビットのデータ長を有し、マークによって指示されるエレメンタリストリームを含んでいるTSパケットのPID(Packet Identification)の値を示す。フィールド Durationは、45kHzのクロックを単位とした計測による、32ビットのデータ長を有する符号無し整数である。このフィールドDurationに格納される値が”0”であれば、このフィールドDurationは、意味を成さない。

図9は、ブロックblkExtensionData()の構造を表すシンタクスを示す図である。フィールドLengthは、32ビットのデータ長を有し、このフィールドLengthの直後からブロックblkExtensionData()の終わりまでのデータ長をバイト数で示す。フィールドDataBlockStartAddressは、32ビットのデータ長を有し、このシンタクス中の、拡張データの本体が格納されるブロックDataBlock()の開始アドレスを、このブロックblkExtensionData()の先頭バイトからの相対バイト数で示す。24ビットのデータ長を有する領域reservedを介してフィールドNumberOfExtDataEntriesが配される。フィールドNumberOfExtDataEntriesは、8ビットのデータ長を有し、このブロックblkExtensionData()のブロックDataBlock()に格納される拡張データのエントリ数を示す。拡張データのエントリは、拡張データの本体を取得するための情報が格納される。フィールドExtDataTypeは、16ビットのデータ長を有し、このブロックblkExtensionData()に記述される拡張データが記録装置用の拡張データであることを表す。このフィールドExtDataTypeの値は、拡張データを識別する第1の値であり、このブロックblkExtensionData()を含むライセンサ(使用認可者)が割り当てるに定義することができる。フィールドExtDataVersionは、拡張データを識別する第2の値であり、この拡張データのバージョン番号を表すものと定義することができる。フィールドExtDataStartAddressは、32ビットのデータ長を有し、このフィールドExtDataStartAddressが含まれる拡張データのエントリ(ブロックext_data_entry())に対応する拡張データの開始アドレスを示す。フィールドExtDataStartAddressは、ブロックblkExtensionData()の先頭バイトからの相対バイト数で、拡張データext_dataの開始アドレスを示す。フィールドExtDataLengthは、32ビットのデータ長を有し、このフィールドExtDataStartAddressが含まれる拡張データのエントリ(ブロックext_data_entries())に対応する拡張データのデータ長を示す。フィールドNumberOfExtDataEntriesで示された個数だけ、拡張データのエントリ(ブロックext_data_entry())が記述されると、それぞれ16ビットのデータ長を有し任意のデータ列からなるフィールドpadding_wordが、2フィールドを組として任意の回数L1だけ繰り返される。その後、拡張データの本体が格納されるブロックDataBlock()が記述される。ブロックDataBlock()は、1以上の拡張データが格納される。それぞれの拡張データext_dataは、上述したフィールドExtDataStartAddressフィールドExtDataLengthに基づき、ブロックDataBlock()から取り出される。

次に、プレイリストファイルに対して定義される拡張データブロックについて説明する。本発明の特定区間にに対する映像の有効性を示す第3の管理情報はプレイリストファイルの

10

20

30

40

50

拡張データブロック内に記述することとする。図10は、プレイリストファイル内のブロックblkExtensionData()におけるブロックDataBlock()（図9）の構造を表すシンタクスを示す図である。この図10の例では、ブロックDataBlock()がブロックblkPlayListExtensionData()として記述されている。先ず、上述の図9を参照し、プレイリストファイル内のブロックblkExtensionData()において、フィールドExtDataTypeおよびフィールドExtDataVersionがそれぞれ所定の値、例えば上述と同様にそれぞれ値”0x1000”、値”0x0100”とする。ブロックblkPlayListExtensionData()において、フィールドTypeIndicatorは、次に続くデータの種類を示す、ISO646に規定された符号化方式で符号化した4文字からなる所定の文字列が記述される。このフィールドTypeIndicatorに記述される文字列によって、次に続くデータ種類がプレイリストファイルにおける拡張データであることが示される。フィールドTypeIndicatorの次に、32ビット(4バイト)のデータ長を有する領域reservedを介して32ビットのデータ長を有するフィールドPlayListMarkExtStartAddressが配され、その次に、32ビットのデータ長を有するフィールドMakersPrivateDataStartPositionが配される。フィールドPlayListMarkExtStartPositionおよびフィールドMakersPrivateDataStartPositionは、それぞれ、ブロックblkPlayListMarkExt()およびブロックblkMakersPrivateData()の、このブロックblkPlayListExtensionData()先頭を基準とした開始アドレスが示される。フィールドMakersPrivateDataStartPositionの次に、192ビットのデータ長を有する領域reservedを介してブロックblkPlayListMarkExtMeta()が配される。16ビットのデータ長を有するパディングワードpadding_wordが値N1で示される回数だけ繰り返され、次に、ブロックblkPlayListMarkExt()が配される。さらに、16ビットのデータ長を有するパディングワードpadding_wordが値N2で示される回数だけ繰り返され、次に、ブロックblkMakersPrivateData()が配される。ブロックblkMakersPrivateData()の後ろには、16ビットのデータ長を有するパディングワードpadding_wordが値N3で示される回数だけ繰り返される。

図11は、ブロックblkPlayListMeta()の構造を表すシンタクスを示す図である。フィールドLengthは、32ビットのデータ長を有し、このフィールドLength直後からこのブロックblkPlayListMeta()の終わりまでのデータ長を示す。フィールドLengthの次に、それぞれ16ビットのデータ長を有するフィールドMakerIDおよびフィールドMakerModelCodeが配される。フィールドMakerIDおよびフィールドMakerModelCodeは、このプレイリストファイルを更新したメーカーおよび当該メーカーの機種を識別する情報がそれぞれ記述される。フィールドMakerModelCodeの次に、32ビットのデータ長を有する領域reservedを介して16ビットのデータ長を有するフィールドRefToMenuThumbnailIndexが配される。このフィールドRefToMenuThumbnailIndexは、このプレイリストファイルにより再生される一連のクリップを代表するサムネイル画像が存在する場合、そのサムネイル画像を特定するサムネイル番号が記述される。次に8ビットのデータ長を有するブロックblkTimeZone()が配され、このプレイリストファイルを更新した際に、装置に設定されているタイムゾーンを示す情報が記述される。さらに56ビットのデータ長を有するフィールドRecordTimeAndDateが配され、このプレイリストファイルを更新した時刻および日付が記述される。フィールドRecordTimeAndDateの次に、8ビットのデータ長を有する領域reservedを介して、それぞれ8ビットのデータ長を有するフィールドPlayListCharacterSetおよびフィールドPlayListNameLengthが配され、さらに続けて、255バイトのデータ長を有する

10

20

30

40

50

フィールド PlayListName が配される。これらフィールド PlayListCharacterSet、フィールド PlayListNameLength およびフィールド PlayListName により、このプレイリストファイルにより示されるプレイリストに付された名前に関する情報が記述される。フィールド PlayListName の次に、ブロック Additional_data() が配される。このブロック Additional_data() は、追加的な情報を格納するために予約された領域であって、32ビットのデータ長を有するフィールド Length2 に示される値のバイト数分のデータ長を有する領域が予約される。

【0014】

図12は、プレイリストファイルにおける拡張データブロック blkPlayListExtensionData() 内のブロック blkPlayListMarkExt() の構造を表すシンタクスを示す図である。フィールド Length は、32ビットのデータ長を有し、このフィールド Length の直後からブロック blkPlayListMarkExt() の最後までのデータ長を示す。フィールド NumberOfPlayListMarks は、16ビットのデータ長を有し、このブロック blkPlayListMarkExt() に含まれる blkMarkExtension() の数を示す。次の for ループ文に従い、フィールド NumberOfPlayListMarks で示される数だけ blkMarkExtension() の情報が記述される。フィールド MakerID およびフィールド MakerModelCode は、それぞれ16ビットのデータ長を有し、メーカーの識別情報および当該メーカーによる機種の識別情報が記述される。フィールド MakerModelCode の次に、29ビットのデータ長を有する領域 reserved を介して 16ビットのデータ長を有するフィールド Pulldown が配される。フィールド Pulldown、フィールド MarkWriteProtectFlag は、本発明と関連性が薄いので、説明を省略する。フィールド RefToMenuThumbnailIndex は、この MarkExtension を代表するサムネイル画像が存在する場合、そのサムネイル画像を特定するサムネイル番号が記述される。次に 8ビットのデータ長を有するブロック blkTimezone() が配され、このプレイリストファイルを更新した際に、装置に設定されているタイムゾーンを示す情報が記述される。さらに 56ビットのデータ長を有するフィールド RecordTimeAndDate が配され、このプレイリストファイルを更新した時刻および日付が記述される。フィールド RecordTimeAndDate の次に、8ビットのデータ長を有する領域 reserved を介して、それぞれ8ビットのデータ長を有するフィールド MarkCharacterSet およびフィールド MarkNameLength が配され、さらに続けて、24バイトのデータ長を有するフィールド MarkName が配される。これらフィールド MarkCharacterSet、フィールド MarkNameLength およびフィールド MarkName により、MarkExtension に付された名前に関する情報が記述される。フィールド MarkName の次に、128バイトのデータ長を有するフィールド MakersInformation が配される。フィールド MakersInformation はメーカーが自由に記述することが可能なフィールドで、本発明の特定区間にに対する映像の有効性を示す第3の管理情報は、フィールド MakersInformation に記述することとする。フィールド MakersInformation の次に 32バイトのデータ長を有するフィールド blkTimeCode が配される。フィールド blkTimeCode はこの MarkExtension に関連した PlayListMark が参照している映像のタイムコードが記述される。

【0015】

図13は、プレイリストファイルにおけるブロック blkMakersPrivateData() の構造を表すシンタクスを示す図である。ブロック blkMakersPrivateData() は、このプレイリストファイルに関して、メーカー独自の情報が記述されるブロックである。フィールド Length は、32ビットのデータ長を有し、このフィールド Length 直後からこのブロック blkMakersPrivateData

10

20

30

40

50

t a()の終わりまでのデータ長を示す。フィールド DataBlockStartAddress は、32ビットのデータ長を有し、このシンタクス中の、メーカ独自情報本体が格納されるブロック DataBlock()の開始アドレスを、このブロックblkMakersPrivateData()の先頭バイトからの相対バイト数で示す。24ビットのデータ長を有する領域 reserved を介して、8ビットのデータ長を有するフィールド NumberOfMakerEntries が配される。次の for 文に従い、フィールド NumberOfMakerEntries で示される個数だけ拡張データのエントリ、すなわち、フィールド MakerID、フィールド MakerModelCode、フィールド MpdsStartAddress およびフィールド MpdlLength が記述される。フィールド MakerID およびフィールド MakerModelCode は、それぞれ16ビットのデータ長を有し、メーカの識別情報および当該メーカによる機種の識別情報が記述される。また、フィールド MpdsStartAddress およびフィールド MpdlLength は、それぞれ32ビットのデータ長を有し、拡張データの本体が格納されるブロック DataBlock()の、このブロックblkExpansionData()の先頭バイトからの相対バイト数による開始アドレスと、データ長とを示す。フィールド NumberOfMakerEntries で示される個数だけ拡張データのエントリが記述されると、それぞれ16ビットのデータ長を有し任意のデータ列からなるフィールド padding_word が、2フィールドを組として任意の回数 L 1だけ繰り返される。その後、拡張データの本体が格納されるブロック DataBlock()が記述される。ブロック DataBlock()は、1以上の拡張データ ext_data が格納される。すなわち、フィールド MakerID およびフィールド MakerModelCode で示されるメーカおよび機種毎に、メーカ独自の拡張データがブロック DataBlock()に格納される。それぞれの拡張データは、上述したフィールド MpdsStartAddress およびフィールド MpdlLength に基づき、ブロック DataBlock()から取り出される。

【0016】

図14は、本発明の記録装置の構成を概略的に示す図である。指示入力部1401はユーザから撮影開始および撮影停止の指示を受けると、その情報を解析部1402に送信する。信号変換部1403は、パン、ズーム、手ぶれ、ピンボケなどの機器情報およびセンサーから受信した情報（たとえば地面の撮影）を解析部1402に送信する。解析部1402は、所定のアルゴリズムにより、映像の有効性、無効性を判定し、重要度を表す値を算出する。重要度を算出するアルゴリズムとしては、たとえばWO2008/032739にて開示されている方法がある。ビデオエンコーダ1404は、ビデオカメラに入力される映像を所定の圧縮符号化方式、例えばH.264|AVCにより圧縮符号化する。ビデオエンコーダ1404で圧縮符号化されたビデオデータは、H.264|AVCのエレメンタリストリーム(ES)として多重化部1406に供給される。ここで、映像の圧縮符号化方式は、H.264|AVCに限られるものではない。オーディオエンコーダ1405は、ビデオカメラに入力される音声を所定の圧縮符号化方式、例えばAC3により圧縮符号化する。オーディオデータの圧縮符号化方式は、AC3に限られるものではない。オーディオデータを圧縮符号化せず、ベースバンドのデータのまま用いることも考えられる。圧縮符号化されたオーディオデータは、多重化部1406に供給される。多重化部1406は、それぞれ圧縮符号化されて供給されたビデオデータおよびオーディオデータを所定の方式で多重化し、1つのAVストリームとして記録部1407に供給する。管理情報生成部1408は、解析部1402および多重化部1406から供給された情報を基に、上述したインデックスファイル、ムービーオブジェクトファイル、BD-Jオブジェクトファイル、プレイリストファイルおよびクリップインフォメーションファイルに格納するための情報を生成する。アプリケーションモジュール1409は、所定の再生条件によりダイジェスト再生を実行するJAVA(登録商標)の実行モジュールで記録部1407に供給される。記録部1407は、多重化部1406、管理情報生成部1408、アプリケーションモジュール1409からのデータをインデックスファイル、ムービーオブジ

10

20

30

40

50

エクトファイル、B D - J オブジェクトファイル、プレイリストファイルおよびクリップインフォメーションファイルとして記録媒体 1 0 3 に対する記録を制御する。

図 1 5 は、本発明の管理情報と A V ストリームのデータ構造を示す図である。プレイリストファイルは本発明の第 2 の管理情報が記述されており、図 1 5 のプレイリストには 3 個のプレイアイテムの再生順序が定義されている。プレイアイテムのそれぞれは、クリップインフォメーションファイルを参照する。クリップインフォメーションファイルは本発明の第 1 の管理情報であり、クリップの実体であるクリップ A V ストリームファイルと 1 対 1 の関係にある。プレイリストファイルに対して記録開始位置には、エントリマークが設けられる。図 1 5 では、3 回記録を実施したことがわかる。また、図 1 5 には 5 個のリンクポイントが設けられている。リンクポイント及び拡張情報は本発明の第 3 の管理情報である。すなわち、このプレイリストには、5 個の重要度情報が含まれており、リンクポイントに対応するプレイリストファイルの拡張データブロックの情報には、それぞれの区間と区間の重要度の情報が記録されている。

図 1 6 は、プレイリストファイルにおけるブロック b l k P l a y L i s t M a r k E x t () の M a k e r s I n f o r m a t i o n に定義した本発明の第 3 の管理情報の構造を表すシンタクスを示す図である。フィールド T y p e は 8 ビットのデータ長を有し、情報の種類を示す値である。フィールド S c o r e は 8 ビットのデータ長を有し、本情報示す区間の重要度を示す値である。フィールド I N T i m e 、フィールド O U T T i m e は区間の開始点と終了点で、それぞれ 4 5 k H z のクロックを単位とした計測による、3 2 ビットのデータ長を有する符号無し整数である。

これら第 3 の管理情報は、アプリケーションモジュール内のプログラムから参照することが可能であり。プログラムが解釈可能なフォーマットとして上記領域に記録する。

以下では、本発明によって、前述したように生成した記録媒体を利用して動画を再生する再生装置について詳細に説明する。図 1 7 は、本発明の再生装置のブロック図である。

読み取り部 1 7 0 1 は記録媒体 1 0 3 より図 4 で説明したファイルの読み込みを行う。

U O コントローラ 1 7 0 2 は、リモコンや再生装置のフロントパネルに対してなされたユーザ操作を検出して、ユーザ操作を示す情報（以降 U O (U s e r O p e r a t i o n) という）をモジュールマネージャ 1 7 0 3 に出力する。モジュールマネージャ 1 7 0 3 は、初期実行及び U O あるいはナビゲーションの変更により生じたタイトル変更を処理するようにナビゲーションエンジン 1 7 0 4 を制御する。ナビゲーションエンジン 1 7 0 4 は、管理情報 2 0 2 の内容によってプレゼンテーションエンジン 1 7 0 5 により動画の再生を制御する。プレゼンテーションエンジン 1 7 0 5 は A V ストリーム 2 1 0 をデコードして画面に出力する。

プログラムエンジン 1 7 0 6 は読み取り部 1 7 0 1 よりロードしたアプリケーション 2 0 3 を実行する。アプリケーション 2 0 3 は、規定の A P I によりプレゼンテーションエンジン 1 7 0 5 及びナビゲーション 1 7 0 4 を制御することができる。また、プログラムエンジン 1 7 0 6 はモジュールマネージャ 1 7 0 3 により制御され、ユーザの操作内容により、ダイジェスト区間のみの再生、任意の時間を指定した再生、撮影に失敗した区間以外の再生など、再生経路をインタラクティブに指定した再生を行うことができる。ユーザの操作画面は、プログラムエンジン 1 7 0 6 により出力され、プレンダ 1 7 1 0 によりオーバーレイされて一つの画面として出力される。

【産業上の利用可能性】

【0 0 1 7】

本発明に係る記録媒体は、撮影した映像を手軽に扱うことを可能にし、より付加価値が高い映像の記録装置を市場に供給することができる。故に本発明に係る記録媒体、記録装置は民生機器産業において高い利用可能性をもつ。

【図面の簡単な説明】

【0 0 1 8】

【図 1】本発明に係る記録媒体の、使用行為についての形態を示す図

【図 2】本発明の記録媒体に記録されたデータの種類を示す図

10

20

30

40

50

【図3】図2に示した記録媒体103に記録されたデータの構造を説明するための図

【図4】本発明の記録媒体103に記録されたファイルのディレクトリ構造を説明するための図

【図5】プレイリストファイルの構造を表すシンタクスを示す図

【図6】ブロックblkPlayList()の構造を表すシンタクスを示す図

【図7】ブロックblkPlayItem()の構造を表すシンタクスを示す図

【図8】ブロックblkPlayItem()の構造を表すシンタクスを示す図

【図9】ブロックblkExtensionData()の構造を表すシンタクスを示す図

【図10】プレイリストファイル内のブロックblkExtensionData()におけるブロックDataBlock()（図9）の構造を表すシンタクスを示す図

【図11】ブロックblkPlayListMeta()の構造を表すシンタクスを示す図

【図12】プレイリストファイルにおける拡張データブロックblkPlayListExtExtensionData()内のブロックblkPlayListMarkExt()の構造を表すシンタクスを示す図

【図13】プレイリストファイルにおけるブロックblkMarkersPrivateData()の構造を表すシンタクスを示す図

【図14】本発明の記録装置の構成を概略的に示す図

【図15】本発明の管理情報とAVストリームのデータ構造を示す図

20

【図16】第3の管理情報の構造を表すシンタクスを示す図

【図17】本発明の再生装置のブロック図

【符号の説明】

【0019】

101 ビデオカメラ

30

102 撮影対象

103 記録媒体

104 再生装置

105 リモコン

106 テレビ

201 AVストリーム

202 管理情報

203 アプリケーション

301 クリップAVストリーム

302 クリップ情報

303 ムービープレイリスト

304 プレイアイテム

305 ムービーオブジェクト

306 BD-Jオブジェクト

307 インデックステーブル

310 プレイリストエクステンション

40

1401 指示入力部

1402 解析部

1403 信号変換部

1404 ビデオエンコーダ

1405 オーディオエンコーダ

1406 多重化部

1407 記録部

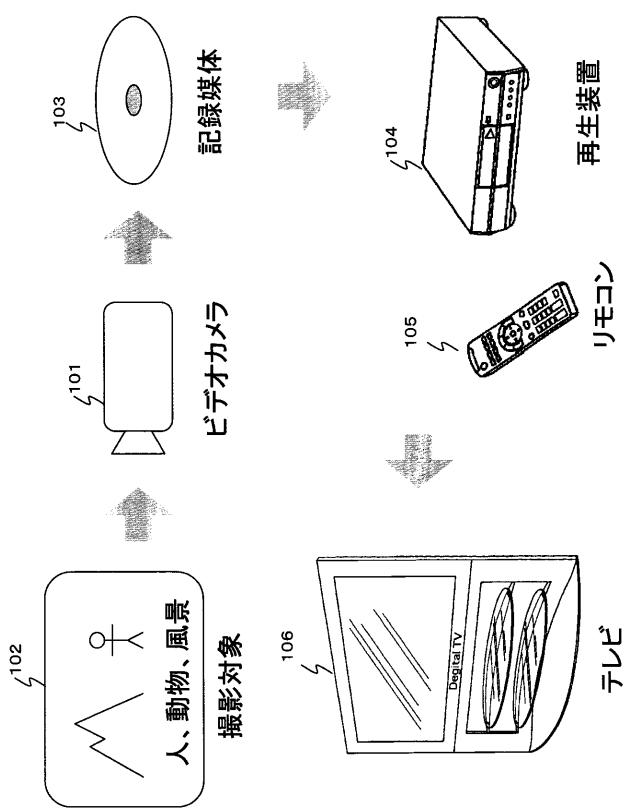
1408 管理情報生成部

1409 アプリケーションモジュール

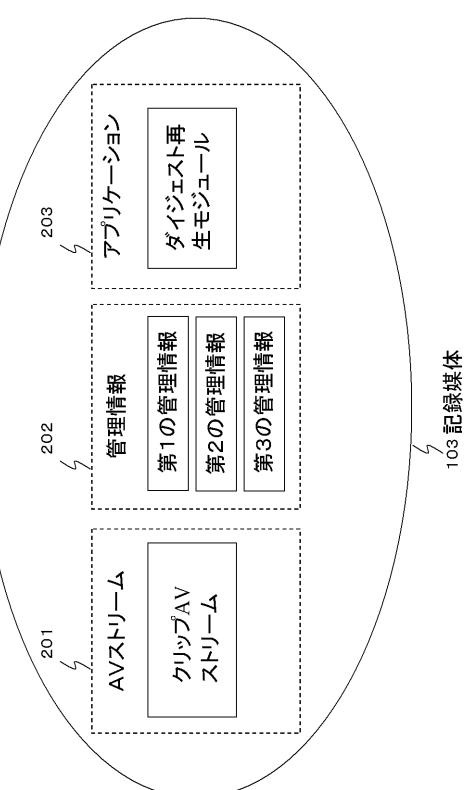
50

- | | |
|---------|---------------|
| 1 7 0 1 | 読み取り部 |
| 1 7 0 2 | UOコントローラ |
| 1 7 0 3 | モジュールマネージャ |
| 1 7 0 4 | ナビゲーションエンジン |
| 1 7 0 5 | プレゼンテーションエンジン |
| 1 7 0 6 | プログラムエンジン |
| 1 7 1 0 | ブレンダ |

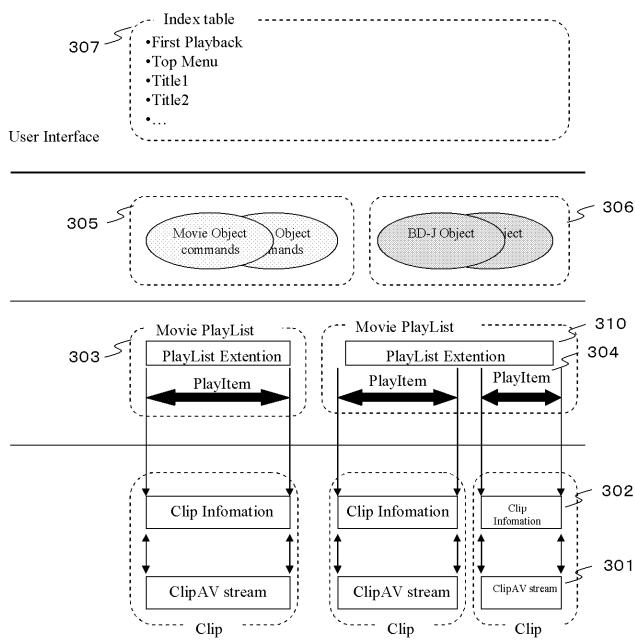
【図1】



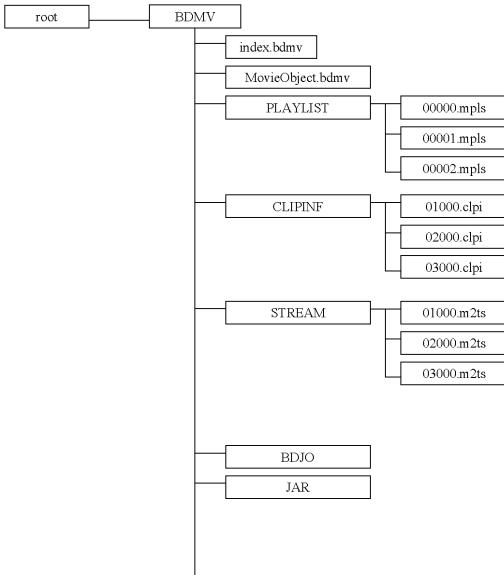
【図2】



【図3】



【図4】



【図5】

シンタクス	データ長 (ビット)	ニーモニック
Movie PlayList file {		
TypeIndicator	8 * 4	bslbf
TypeIndicator2	8 * 4	bslbf
PlayListStartAddress	32	uimsbf
PlayListMarkStartAddress	32	uimsbf
ExtensionDataStartAddress	32	uimsbf
reserved	160	bslbf
blkAppInfoPlayList()		
for (i=0; i<N1; i++) {		
padding_word	16	bslbf
}		
blkPlayList()		
for (i=0; i<N2; i++) {		
padding_word	16	bslbf
}		
blkPlayListMark()		
for (i=0; i<N3; i++) {		
padding_word	16	bslbf
}		
blkExtensionData()		
for (i=0; i<N4; i++) {		
padding_word	16	bslbf
}		

【図6】

シンタクス	データ長 (ビット)	ニーモニック
blkPlayList()		
Length	32	uimsbf
reserved	16	bslbf
NumberOfPlayItems	16	uimsbf
NumberOfSubPaths	16	uimsbf
for(PlayItem_id=0;		
PlayItem_id<NumberOfPlayItems; PlayItem_id++) {		
blkPlayItem()		
}		
for(SubPath_id=0;		
SubPath_id<NumberOfSubPaths; SubPath_id++) {		
blkSubPath()		
}		

【図7】

シンタクス		データ長 (ビット)	ニーモニック
blkExtensionData()			
Length	32	uimsbf	
if(Length != 0){			
DataBlockStartAddress	32	uimsbf	
reserved	24	uimsbf	
NumberOfExtDataEntries	8	uimsbf	
for (i=0; i<NumberOfExtDataEntries; i++){			
ext_data_entry{			
ExtData Type	16	uimsbf	
ExtDataVersion	16	uimsbf	
ExtDataStartAddress	32	uimsbf	
ExtDataLength	32	uimsbf	
}			
for (i=0; i<1; i++){			
padding_word	16	bslbf	
padding_word	16	bslbf	
 }			
DataBlock()			
	32 + 8 * (Length - DataBlockStartAddress)		
}			
}			

【図8】

シンタクス		データ長 (ビット)	ニーモニック
blkPlaylistMark()			
Length	16	uimsbf	
ClipInformationFileName	8 * 5	bslbf	
ClipCodeIdentifier	8 * 4	bslbf	
reserved	12	bslbf	
ConnectionCondition	4	bslbf	
RetroSTCID	8	uimsbf	
INTime	32	uimsbf	
OUTTime	32	uimsbf	
blkUOMaskTable()			
PlayItemRandomAccessFlag	1	bslbf	
reserved	7	bslbf	
StillMode	8	bslbf	
if (StillMode == 0x01){			
StillTime	16	uimsbf	
} else {			
reserved	16	bslbf	
blkSNTTable()			
}			

【図9】

シンタクス		データ長 (ビット)	ニーモニック
blkPlaylistExtensionData()			
TypeIndicator	8 * 4	bslbf	
reserved	8 * 4	bslbf	
PlaylistMarkExtStartAddress	32	uimsbf	
MakersPrivateDataStartAddress	32	uimsbf	
reserved	192	bslbf	
blkPlaylistMeta()			
for (i=0; i<N1; i++){			
padding_word	16	bslbf	
}			
blkPlaylistMarkExt()			
for (i=0; i<N2; i++){			
padding_word	16	bslbf	
}			
blkMakersPrivateData()			
for (i=0; i<N3; i++){			
padding_word	16	bslbf	
}			

【図10】

シンタクス		データ長 (ビット)	ニーモニック
blkPlaylistExtensionData()			
Length	32	uimsbf	
NumberOfPlayListMarks	16	uimsbf	
for(PL_mark_id=0; PL_mark_id < NumberOfPlayListMarks; PL_mark_id++) {			
reserved	8	bslbf	
MarkType	8	bslbf	
RefToPlayItemID	16	uimsbf	
MarkTimeStamp	32	uimsbf	
EntryESPID	16	uimsbf	
Duration	32	uimsbf	
}			

【図 1 1】

シントクス	データ量 (ビット)	シントクス	データ量 (ビット)
blkPlayListMarkExt() {		blkPlayListMeta() {	
Length	32	Length	32
MarkerID	16	MarkerModelCode	16
MarkerModelCode	16	reserved	32
reserved	bslbf	RefToMenuThumbnailIndex	16
blkTimeZone()	29	blkTimeZone()	8
RecordTimeAndDate	16	RecordTimeAndDate	4 * 14
reserved	bslbf	reserved	8
PlayListCharacterSet	8	PlayListCharacterSet	8
PlayListNameLength	8	PlayListNameLength	8
PlayListName	8 * 255	PlayListName	8 * 255
Additional_Data(){		Additional_Data(){	
Length2	32	Length2	32
reserved	bslbf	reserved	bslbf
}		}	
}			

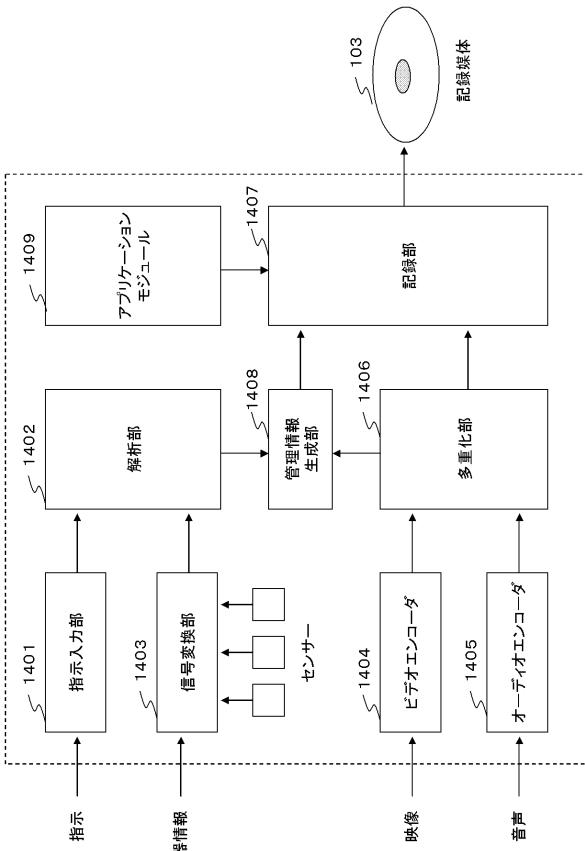
【図 1 2】

シントクス	データ量 (ビット)	シントクス	データ量 (ビット)
blkMakersPrivateData() {		blkMakersPrivateData() {	
Length	32	Length	32
if(Length == 0){	ulmsbf	DataBlockStartAddress	32
DataBlockStartAddress	ulmsbf	reserved	bslbf
NumberOfMarkerEntries	24	NumberOfMarkerEntries	8
for (i=0; i<NumberOfMarkerEntries; i++){	ulmsbf	MarkerID	16
MarkerID	ulmsbf	MarkerModelCode	16
MarkerModelCode	ulmsbf	MpStartAddress	32
MpStartAddress	ulmsbf	MpLength	32
MpLength	ulmsbf	}	
}	ulmsbf	for (i=0; i<Length; i++){	
	ulmsbf	padding_word	16
	ulmsbf	padding_word	16
	ulmsbf	}	
		DataBlock()	32 +
		8 * (Length -	
		DataBlockStartAddress)	
		}	
}		}	

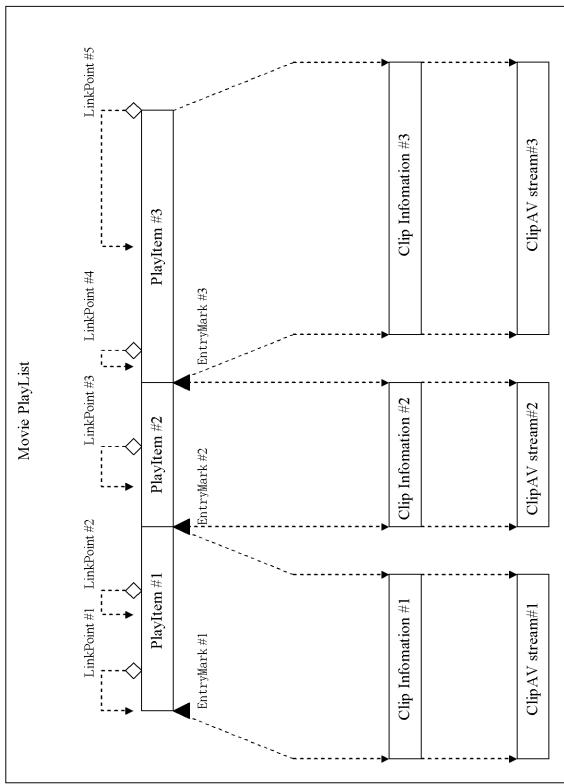
【図 1 3】

シントクス	データ量 (ビット)	シントクス	データ量 (ビット)
blkPlayListStMarkExt() {		NumberOfPlayListMarks	16
Length	32	for(PL_mark_id=0;	ulmsbf
NumberOfPlayListMarks	16	PL_mark_id < NumberOfPlayListMarks; PL_mark_id++) {	ulmsbf
for(PL_mark_id=0;	16	blkMarkExtension() {	ulmsbf
PL_mark_id < NumberOfPlayListMarks; PL_mark_id++) {	16	MarkerID	16
blkMarkExtension() {	16	MarkerModelCode	16
MarkerModelCode	16	reserved	bslbf
reserved	bslbf	blkTimeZone()	16
Pulldown	29	blkTimeZone()	2
MarkerWriteProtectFlag	2	MarkerWriteProtectFlag	1
RefToMarkThumbnailIndex	1	blkTimeZone()	16
blkTimeZone()	16	RecordTimeAndDate	16
RecordTimeAndDate	4 * 14	reserved	bslbf
MarkerCharacterSet	8	PlayListCharacterSet	8
MarkerSetNameLength	8	PlayListNameLength	8
MarkerName	8 * 24	PlayListName	8 * 255
MarkersInformation	128	Additional_Data(){	ulmsbf
blkTimeCode()	32	Length2	32
reserved	16	reserved	8 * Length2
}		}	
}		}	
}			

【図 1 4】



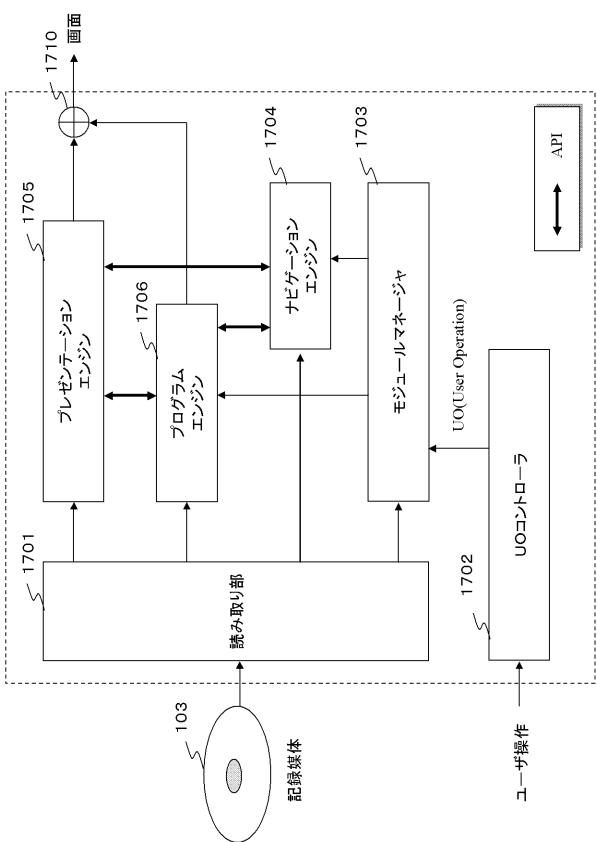
【図15】



【図16】

シンタックス	データ長	ニーモニック
blkMakersInformation()		
Type	8	uimsbf
Score	8	uimsbf
INTime	32	uimsbf
OUTTime	32	uimsbf
reserved	48	bslbf

【図17】



フロントページの続き

(51) Int.Cl.	F I G 1 1 B 27/00	テーマコード(参考) D
--------------	----------------------	-----------------

F ターム(参考) 5C052 AA02 DD01 DD08
5C053 FA14 FA24 HA30 KA05 LA01
5D044 AB07 BC02 CC04 DE03 DE48 EF05 FG24
5D110 AA14 AA26 AA28 CA12 CJ14