(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2009-5371
(P2009-5371A)
(43)公開日　平成21年1月8日(2009.1.8)

(51)Int.Cl.		F I			テーマコード（参考）
H04N 5/93	(2006.01)	H04N	5/93	Z	5C053
G11B 20/10	(2006.01)	G11B	20/10	321Z	5C059
G11B 27/034	(2006.01)	G11B	27/034		5D044
H04N 7/26	(2006.01)	H04N	7/13	Z	5D110

審査請求 有　請求項の数 3　ＯＬ　（全 20 頁）

(21)出願番号　特願2008-182638 (P2008-182638)
(22)出願日　平成20年7月14日 (2008.7.14)
(62)分割の表示　特願2005-33375 (P2005-33375)の分割
原出願日　平成14年6月14日 (2002.6.14)
(31)優先権主張番号　特願2001-180952 (P2001-180952)
(32)優先日　平成13年6月15日 (2001.6.15)
(33)優先権主張国　日本国(JP)

(71)出願人　000005049
シャープ株式会社
大阪府大阪市阿倍野区長池町２２番２２号
(74)代理人　100112335
弁理士　藤本　英介
(74)代理人　100101144
弁理士　神田　正義
(74)代理人　100101694
弁理士　宮尾　明茂
(72)発明者　木山　次郎
大阪府大阪市阿倍野区長池町２２番２２号　シャープ株式会社内
(72)発明者　岩野　裕利
大阪府大阪市阿倍野区長池町２２番２２号　シャープ株式会社内

最終頁に続く

(54)【発明の名称】データ記録方法、データ編集方法およびデータ復号方法、並びにその装置、及び記録媒体

(57)【要約】
【課題】　AVストリームを繋ぎ合わせて構成されたAVストリームを再生する際に、再生の乱れが生じるのを、簡単な構成にて防止することが可能なデータ記録方法、データ編集方法、データ復号方法、データ記録装置、及びデータ復号装置を提供する。
【解決手段】　少なくとも映像又は音声の単独再生可能なビデオユニット（ＶＵ）を含む複数のレコードユニット（ＲＵ）で構成されるＡＶストリームと、前記ＡＶストリームを管理するプログラム情報(Movie atom)と、を、記録媒体に記録し、前記プログラム情報にレコードユニット又はＡＶストリーム間の結合箇所を管理する情報（Edit list atom）を有する。ＡＶストリームの復号において、結合箇所管理情報に基づいて、結合箇所では、例えば、復号の一時停止や復号器の切り替えを行う。
【選択図】　図１８

Entry Number	Track duration D(i)	Media time T(i)	Media rate R(i)
#1	10000	0	1
#2	6000	10000	1

【特許請求の範囲】
【請求項１】
少なくとも映像データを含むＡＶストリームを、該ＡＶストリームに関する位置情報を有するプログラム情報に従って復号するデータ復号方法であって、
前記ＡＶストリームと前記プログラム情報は、記録媒体に記録されており、
前記ＡＶストリームは、１個のファイルとして管理され、
前記プログラム情報は、前記ＡＶストリーム内の連続的に再生可能な区間に対応するエントリ情報を含んでおり、
前記方法は、
前記記録媒体から前記ファイルをデコーダに転送するステップと、
前記デコーダにて、前記ファイルの復号を開始するステップと、
前記エントリ情報の切り替わりに基づき、前記ファイル内の結合箇所を判断するステップと、
前記ファイルの復号時に、前記結合箇所において前記ファイルの復号を一時停止するステップと
を有することを特徴とするデータ復号方法。
【請求項２】
請求項１に記載のデータ復号方法の実行に用いるＡＶストリームおよびプログラム情報が記録されているコンピュータ読み取り可能な記録媒体。
【請求項３】
請求項１に記載のデータ復号方法を記録したコンピュータ読み取り可能な記録媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、映像データ、音声データをハードディスク、光ディスク等のランダムアクセス可能な記録媒体に対して記録・再生するデータ記録方法、データ編集方法およびデータ復号方法、及びその装置、及び記録媒体に関するものである。
【背景技術】
【０００２】
ディスクメディアを用いたビデオや音声のディジタル記録再生装置が普及しつつある。テープメディアにはないディスクメディアにおける特徴機能として、非破壊編集機能あるいはノンリニア編集機能と呼ばれるものがある。この機能は、ディスク上に記録したAVストリームを移動あるいはコピーすることなく、AVストリームの任意の区間を任意の順番で再生できる、というもので、AVストリームのどこからどこまでどういう順番で再生するかを示す情報(再生管理情報)を作り、その情報に従って再生することで実現される。
【０００３】
このようなディスクメディアでは、素材を書き換えたりデータを移動することなく、編集を行うことが可能なわけであるが、素材を直接編集したい場合もある。例えば、非破壊編集した結果を、パーソナル・コンピュータ(PC)で扱いやすいように1個のファイルにまとめたいとする。この場合、編集結果で使用されている部分のみを個々のAVストリームから抜き出し結合して1個のファイルにまとめることになる。
【０００４】
次のような場合もある。ディスクの空き容量を増やすため、AVストリーム中の不要な中間部分を削除したいとする。この場合、中間部分の前後を結合することになる。
【０００５】
以上のような場合、AVストリーム同士を繋ぎあわせることになる訳であるが、ビデオデータの符号化方式にMPEGビデオ規格(ISO/IEC 11172-2やISO/IEC 13818-2)を用いた場合、繋ぎ目において再生が乱れるおそれがある。
【０００６】
その理由について、以下に説明する。MPEGビデオ規格では可変長符号化を採用しており

、符号化データを所定のレートで符号化する場合、VBV(Video Buffering Verifier)と呼ばれる仮想デコーダモデルを符号化器の出力に接続して使用し、そのデコードバッファ(VBVバッファ)がオーバーフローもアンダーフローも発生しないように符号化することが規定されている。

【０００７】

このモデルでは、上記の所定レート以下で符号化データがVBVバッファに入力され、そのレートでVBVバッファのデータ占有量が増加する。その一方、フレームあるいはフィールドの復号が終了する毎、一瞬でその符号化データ量分、データ占有量が減少する。

【０００８】

MPEGビデオによる符号化データは、このような増加・減少を繰り返しても、図２２に示すようなVBVバッファがアンダーフローやオーバーフローを起こさないように制御して符号化されなければ、正常な再生は保証されない。ビデオデータを繋ぎ合わせた場合に、再生が乱れるおそれがあるのは、繋ぎ合わせた箇所において、VBVバッファがアンダーフローやオーバーフローしてしまうおそれがあるためである。

【０００９】

繋ぎ合わせた箇所でＶＢＶバッファが破綻する理由について、例を挙げて説明する。ここでは、VBVバッファの占有量の時間遷移がそれぞれ図２３(a) に示すビデオ符号化データAの時間OUTの前半部と、図２３(b)に示すビデオ符号化データBの時間INの後半部とを結合する場合について説明する。

【００１０】

図２３(c)にその結果を示す。結合点直前のバッファ占有量が低いのにも関わらず、結合点直後で符合化データ量の大きいフレームあるいはフィールドの復号が発生しているため、バッファのアンダーフローが発生していることがわかる。ここのようなことが発生するのは、繋ぎ目においてはバッファ占有量の整合が取れない可能性があるためである。

【００１１】

上述の問題を解決する方法として、例えば特開平9-182024号公報では、復号器の入力データの転送速度を高めることにより、アンダーフローを防止する技術が提案されている（従来技術１、例えば、特許文献１参照）。

【００１２】

また別の方法として、例えば特開平8-251582号公報では、結合の際に繋ぎ目の部分を一旦復号し、VBVバッファが破綻しないような符号化データ量になるように符号化し直す技術（再符号化）が提案されている（従来技術２、例えば、特許文献２参照）。

【特許文献１】特開平9-182024号公報

【特許文献２】特開平8-251582号公報

【発明の開示】

【発明が解決しようとする課題】

【００１３】

しかしながら、従来技術１の場合、特殊な復号器を用意する必要があり、コストの面で不利になるという問題点がある。

【００１４】

また、従来技術２の場合においては、再符号化により画質が低下するおそれがあり、さらに符号化と復号を逐次的あるいは並列的に行う必要あるため、装置が複雑化するという問題も含んでいる。

【００１５】

本発明は、上記課題に鑑みてなされたものであり、AVストリームを繋ぎ合わせて構成されたAVストリームを再生する際に、再生の乱れが生じるのを、簡単な構成にて防止することが可能なデータ記録方法、データ編集方法、データ復号方法、データ記録装置、及びデータ復号装置を提供することを目的とする。

【課題を解決するための手段】

【００１６】

本願の第１の発明は、少なくとも映像又は音声からなる第１のデータを含む複数の第１のユニットで構成される第２のユニットと、前記第２のユニットを管理する第１のプログラムとを、記録媒体に記録するデータ記録方法であって、前記第１のプログラムが、前記第１のユニット間の結合箇所を管理する情報を有することを特徴とする。

【００１７】

本願の第２の発明は、前記結合箇所は、前記第２のユニット中から任意の前記第１のデータを削除した箇所であることを特徴とする。

【００１８】

本願の第３の発明は、前記第２のユニットは、１個のファイルで管理されることを特徴とする。

【００１９】

本願の第４の発明は、前記映像の第１のデータは、該映像をＭＰＥＧ規格により符号化したものであることを特徴とする。

【００２０】

本願の第５の発明は、少なくとも映像又は音声からなる第１のデータを含む第１のユニットと、少なくとも映像又は音声からなる第２のデータを含む第３のユニットとを接続して、第２のユニットを作成するデータ編集方法であって、前記第２のユニットを管理する第１のプログラムが、前記第１のユニットと前記第３のユニットとの結合箇所を管理する情報を有することを特徴とする。

【００２１】

本願の第６の発明は、前記第１のユニットと前記第３のユニットは、前記第２のユニット中の任意の前記第１のデータを削除することによって形成されることを特徴とする。

【００２２】

本願の第７の発明は、前記第２のユニットは、１個のファイルで管理されることを特徴とする。

【００２３】

本願の第８の発明は、前記映像の第１、第２のデータは、該映像をＭＰＥＧ規格により符号化したものであることを特徴とする。

【００２４】

本願の第９の発明は、少なくとも映像又は音声からなる第１のデータを含む複数の第１のユニットで構成される第２のユニットを、該第２のユニットを管理する第１のプログラムに従って復号するデータ復号方法であって、前記第１のプログラムが、前記第１のユニット間の結合箇所を管理する情報を有し、前記結合箇所情報に基づき、前記第１のユニットの復号を制御することを特徴とする。

【００２５】

本願の第１０の発明は、前記第１のユニットの復号の制御は、前記結合箇所において復号を一時停止することを特徴とする。

【００２６】

本願の第１１の発明は、前記第１のユニットの復号の制御は、前記結合箇所の前後で復号器を切り替えることを特徴とする。

【００２７】

本願の第１２の発明は、前記映像の第１，第２のデータは、該映像をＭＰＥＧ規格により符号化したものであることを特徴とする。

【００２８】

本願の第１３の発明は、少なくとも映像又は音声からなる第１のデータを含む複数の第１のユニットで構成される第２のユニットと、前記第２のユニットを管理する第１のプログラムとを、記録媒体に記録するデータ記録装置であって、前記第１のプログラムが、前記第１のユニット間の結合箇所を管理する情報を有することを特徴とする。

【００２９】

本願の第１４の発明は、少なくとも映像又は音声からなる第１のデータを含む第１のユ

ニットと、少なくとも映像又は音声からなる第２のデータを含む第３のユニットとを接続して、第２のユニットを作成するデータ編集装置であって、前記第２のユニットを管理する第１のプログラムが、前記第１のユニットと第３のユニットとの結合箇所を管理する情報を有することを特徴とする。

【００３０】

本願の第１５の発明は、少なくとも映像又は音声からなる第１のデータを含む複数の第１のユニットで構成される第２ユニットを、該第２のユニットを管理する第１のプログラムに従って復号するデータ復号装置であって、前記第１のプログラムが、前記第１のユニットの結合箇所情報を管理し、前記結合箇所情報に基づき、復号の制御を行う前記第１のユニットの復号器を備えたことを特徴とする。

【発明を実施するための最良の形態】

【００３１】

以下、本発明の実施形態について、図面を参照しながら詳細に説明する。

＜システム構成＞

図１は本実施形態において共通に用いる、アフレコ可能なビデオディスクレコーダの構成である。この装置は、図１に示すように、バス100、ホストCPU101、RAM102、ROM103、ユーザインタフェース104、システムクロック105、光ディスク106、ピックアップ107、ECCデコーダ108、ECCエンコーダ109、再生用バッファ110、記録/アフレコ用バッファ111、デマルチプレクサ112、マルチプレクサ113、多重化用バッファ114、オーディオデコーダ115、ビデオデコーダ116、オーディオエンコーダ117、ビデオエンコーダ118および図示されないカメラ、マイク、スピーカ、ディスプレイ等で構成される。

【００３２】

ホストCPU101は、バス100を通じてデマルチプレクサ112、マルチプレクサ113、ピックアップ107、また図示していないが、オーディオデコーダ115、ビデオデコーダ116、オーディオエンコーダ117、ビデオエンコーダ118との通信を行う。

【００３３】

再生時には、光ディスク106からピックアップ107を通じて読み出されたデータは、ECCデコーダ108によって誤り訂正され、再生用バッファ110に一旦蓄えられる。デマルチプレクサ112は、オーディオデコーダ115、ビデオデコーダ116からのデータ送信要求に従って、再生用バッファ中のデータをその種別によって適当なデコーダに振り分ける。

【００３４】

一方、記録時には、オーディオエンコーダ117とビデオエンコーダ118によって圧縮符号化されたデータは、多重化用バッファ114に一旦送られ、マルチプレクサ113によってAV多重化され、記録/アフレコ用バッファ111に送られる。記録/アフレコ用バッファ111中のデータは、ECCエンコーダ109によって誤り訂正符号を付加され、ピックアップ107を通じて光ディスク106に記録される。

【００３５】

オーディオデータの符号化方式にはMPEG-1 Layer-IIを、ビデオデータの符号化方式にはMPEG-2をそれぞれ用いる。

【００３６】

光ディスク106は、外周から内周に向かって螺旋状に記録再生の行われる脱着可能な光ディスクとする。2048byteを1セクタとし、誤り訂正のため16セクタでECCブロックを構成する。ECCブロック中のデータを書き換える場合、そのデータが含まれるECCブロック全体を読み込み、誤り訂正を行い、対象のデータを書き換え、再び誤り訂正符号を付加し、ECCブロックを構成し記録媒体に記録する必要がある。

【００３７】

また、記録効率を上げるためZCAV(ゾーン角速度一定)を採用しており、記録領域は回転数の異なる複数のゾーンで構成される。

【００３８】

＜ファイルシステム＞

光ディスク106上の各種情報を管理するためにファイルシステムを用いる。ファイルシステムには、PCとの相互運用を考慮してUDF(Universal Disk Format)を使用する。ファイルシステム上では、各種管理情報やAVストリームはファイルとして扱われる。ユーザエリアは、2048byteの論理ブロック(セクタと一対一対応)で管理される。

【００３９】

各ファイルは、整数個のエクステント(連続した論理ブロック)で構成され、エクステント単位で分散して記録してもよい。空き領域は、Space Bitmapを用いて論理ブロック単位で管理される。

【００４０】

上記エクステントを表す情報やSpace Bitmap等、ファイルシステムに関する管理情報は光ディスク106 上に記録される。

【００４１】

＜ファイルフォーマット＞

AVストリーム管理のためのフォーマットとして、QuickTimeファイルフォーマットを用いる。QuickTimeファイルフォーマットとは、Apple社が開発したマルチメディアデータ管理用フォーマットであり、PCの世界で広く用いられている。

【００４２】

QuickTimeファイルフォーマットは、ビデオデータやオーディオデータ等(これらを総称してメディアデータとも呼ぶ)と管理情報とで構成される。両者を合わせてここでは、QuickTimeムービー(略してムービー)と呼ぶ。両者は同じファイル中に存在しても、別々のファイルに存在してもよい。

【００４３】

同じファイル中に存在する場合は、図２(a)に示すファイル201のような構成をとる。各種情報は、atomという共通の構造に格納される。管理情報はMovie atomという構造に格納され、AVストリームはMovie data atomという構造に格納される。

【００４４】

尚、Movie atomはメディアデータの任意の区間を任意の順序で再生することを制御するための一種のプログラム情報であり、メディアデータ中の任意の時間に対応するAVデータのファイル中での相対位置を導くためのテーブルや、メディアデータの属性情報や、後述する外部参照情報などが含まれている。

【００４５】

一方、管理情報とメディアデータを別々のファイルに格納した場合は、図２(b)に示すファイル202,203のような構成をとる。管理情報はMovie atomという構造に格納されるが、AVストリームはatomには格納される必要はない。このとき、Movie atomはAVストリームを格納したファイル203を「外部参照」している、という。

【００４６】

外部参照は、図２(c)に示すように、ファイル204のMovie atom が複数のファイル205,206 に設けたAVストリームファイル#1,#2に対して行うことが可能であり、この仕組みにより、AVストリーム自体を物理的に移動することなく、見かけ上編集を行ったように見せる、いわゆる「ノンリニア編集」「非破壊編集」が可能になる。

【００４７】

それでは、図３乃至図１２を用いて、QuickTimeの管理情報のフォーマットについて説明する。まず、共通の情報格納フォーマットであるatomについて説明する。atomの先頭には、そのatomのサイズであるAtom size、そのatomの種別情報であるTypeが必ず存在する。Typeは4文字で区別され、例えばMovie atomでは ’moov’、Movie data atomでは ’mdat’となっている。

【００４８】

各atomは別のatomを含むことができる。すなわち、atom間には階層構造がある。Movie atomの構成を図３に示す。Movie header atom は、そのMovie atomが管理するムービーの全体的な属性を管理する。Track atomは、そのムービーに含まれるビデオやオーディオ等

のトラックに関する情報を格納する。User-defined data atomは、独自に定義可能なatomである。

【００４９】

Track atomの構成を図４に示す。Track header atomは、そのトラックの全体的な属性を管理する。Edit atomは、メディアデータのどの区間をムービーのどのタイミングで再生するかを管理する。Track reference atomは、別のトラックとの関係を管理する。Media atomは、実際のビデオやオーディオといったデータを管理する。

【００５０】

Track header atomの構成を図５に示す。ここでは、後での説明に必要なもののみ説明する。flagsは属性を示すフラグの集合である。代表的なものとして、Track enabledフラグがあり、このフラグが1であれば、そのトラックは再生され、0であれば、再生されない。layerはそのトラックの空間的な優先度を表しており、画像を表示するトラックが複数あれば、layerの値が小さいトラックほど画像が前面に表示される。

【００５１】

Media atomの構成を図６に示す。Media header atomは、そのMedia atomの管理するメディアデータに関する全体的な属性等を管理する。Handler reference atomは、メディアデータをどのデコーダでデコードするかを示す情報を格納する。Media information atomは、ビデオやオーディオ等メディア固有の属性情報を管理する。

【００５２】

Media information atomの構成を図７に示す。Media information header atomは、ビデオやオーディオ等メディア固有の属性情報を管理する。Handler reference atomは、Media atomの項で説明したとおりである。Data information atomは、そのQuickTimeムービーが参照するメディアデータを含むファイルの名前を管理するatomであるData reference atomを含む。Sample table atomは、データのサイズや再生時間等を管理している。

【００５３】

次に、Sample table atomについて説明するが、その前に、QuickTimeにおけるデータの管理方法について、図８に示すファイル801を用いて説明する。QuickTimeでは、データの最小単位(例えばビデオフレーム)をサンプル(sample)と呼ぶ。個々のトラック毎に、サンプルには再生時間順に1から番号(サンプル番号)がついている。

【００５４】

QuickTimeフォーマットでは、個々のサンプルの再生時間長およびデータサイズを管理している。また、同一トラックに属するサンプルが再生時間順にファイル中で連続的に配置された領域をチャンク(chunk)と呼ぶ。チャンクにも再生時間順に、1から番号がついている。

【００５５】

また、QuickTimeフォーマットでは、個々のチャンクのファイル先頭からのアドレスおよび個々のチャンクが含むサンプル数を管理している。これらの情報に基づき、任意の時間に対応するサンプルの位置を求めることが可能となっている。

【００５６】

Sample table atomの構成を図９に示す。Sample description atomは、個々のチャンクのデータフォーマット(Data format)やサンプルが格納されているファイルのインデックス等を管理するテーブルであり、データフォーマットやサンプルの格納されているファイル等が異なる場合、テーブル中の異なるエントリで管理される。なお、エントリとは、テーブル中において、他の情報から参照される情報の単位のことである。

【００５７】

Time-to-sample atomは、個々のサンプルの再生時間を管理する。Sync sample atomは、個々のサンプルのうち、デコード開始可能なサンプルを管理する。Sample-to-chunk atomは、個々のチャンクに含まれるサンプル数および個々のチャンクがSample description atom中のどのエントリを参照しているかを管理する。Sample size atomは、個々のサンプルのサイズを管理する。Chunk offset atomは、個々のチャンクのファイル先頭からの

アドレスを管理する。

【００５８】

Edit atomは、図１０に示すように、1個のEdit list atomを含む。Edit list atomは、Number of entriesで指定される個数分の、Track duration、Media time、Media rateの値の組(エントリ)を持つ。各エントリは、トラック上で連続的に再生される区間に対応し、そのトラック上での再生時間順に並んでいる。

【００５９】

Track durationはそのエントリが管理する区間のトラック上での再生時間、Media timeはそのエントリが管理する区間の先頭に対応するメディアデータ上での位置、Media rateはそのエントリが管理する区間の再生スピードを表す。尚、Media timeが-1の場合は、そのエントリのTrack duration分、そのトラックでのサンプルの再生を停止する。この区間のことをempty editと呼ぶ。

【００６０】

図１１にEdit listの使用例を示す。ここでは、Edit list atomの内容が図１１(a)に示す内容であり、さらにサンプルの構成が図１１(b)であったとする。尚、ここではi番目のエントリのTrack durationをD(i)、Media timeをT(i)、Media rateをR(i)とする。このとき、実際のサンプルの再生は図１１(c)に示す順に行われる。このことについて簡単に説明する。

【００６１】

まず、エントリ#1はTrack durationが13000、Media timeが20000、Media rateが1であるため（図１１(a)）、そのトラックの先頭から13000の区間（図１１(c)）はサンプル中の時刻20000から33000(=20000+13000)の区間を再生する（図１１(b)）。次に、エントリ#2はTrack durationが5000、Media timeが-1であるため（図１１(a)）、トラック中の時刻13000から18000(=13000+5000)の区間、何も再生を行わない（null，図１１(c)）。

【００６２】

最後に、エントリ#3はTrack durationが10000、Media timeが0、Media rateが1であるため（図１１(a)）、トラック中の時刻18000(=13000+5000)から28000(=18000+10000)の区間において（図１１(c)）、サンプル中の時刻0から10000の区間を再生する（図１１(b)）。

【００６３】

図１２にUser-defined data atomの構成を示す。このatomには、QuickTimeフォーマットで定義されてない独自の情報を任意個数格納することができる。1個の独自情報は1個のエントリで管理され、1個のエントリはSizeとTypeとUser dataとで構成される。Sizeはそのエントリ自体のサイズを表し、Typeは独自情報をそれぞれ区別するための識別情報、User dataは実際のデータを表す。

【００６４】

＜インデックス・ファイル＞

ディスク内に含まれるQuickTimeムービーを管理するため、AVインデックス・ファイルという特別のQuickTimeムービーファイルをディスク内に１個置く。AVインデックス・ファイルには、ディスク内のファイル(QuickTimeムービーやQuickTimeムービーから参照されている静止画等)に関するサムネイルや各種属性が登録されている。

【００６５】

各種属性の中には、そのファイルが外部参照されている回数を示すlink countがある。link countを参照することで、そのファイルを参照しているファイルがあるかどうかを容易に知ることができ、他から参照されているファイルを不用意に削除してしまうことを防ぐことができる。

【００６６】

＜実施例１＞

本発明の一実施例について、図１３乃至図２１を用いて説明する。

＜AVストリームの形態＞

本実施例におけるAVストリームの構成を、図１３及び図１４とともに説明する。AVストリームは、整数個のRecord Unit(RU)で構成される。RUはディスク上で連続的に記録する単位である。RUの長さは、AVストリームを構成するRUをどのようにディスク上に配置してもシームレス再生(再生中に絵や音が途切れないで再生できること)が保証されるように設定される。この設定方法については後述する。

【００６７】

また、RU境界がECCブロック境界と一致するようにストリームを構成する。RUのこれらの性質によって、AVストリームをディスクに記録した後も、シームレス再生を保証したまま、ディスク上でRU単位の配置を容易に変更することができる。

【００６８】

RUは整数個のVideo Unit(VU)で構成する。VUは単独再生可能な単位であり、そのことから再生の際のエントリ・ポイントとなりうる。VUの構成を図１４に示す。VUは1秒程度のビデオデータを格納した整数個のGOP(グループ・オブ・ピクチャ)とそれらと同じ時間に再生されるメインオーディオデータを格納した整数個のAAU(オーディオ・アクセス・ユニット)で構成される。

【００６９】

尚、GOPはMPEG-2ビデオ規格における圧縮の単位であり、複数のビデオフレーム(典型的には15フレーム程度)で構成される。AAUはMPEG-1 LayerII規格における圧縮の単位で、1152点の音波形サンプル点により構成される。サンプリング周波数が48kHzの場合、AAUあたりの再生時間は0.024秒となる。

【００７０】

VU中ではAV同期再生のために必要となる遅延を小さくするため、AAU、GOPの順に配置する。また、VU単位で独立再生可能なように、VU中のビデオデータの先頭にはSequence Header(SH)を置く。

【００７１】

VUの再生時間は、VUに含まれるビデオフレーム数にビデオフレーム周期をかけたものと定義する。また、VUを整数個組み合わせてRUを構成する場合、RUの始終端をECCブロック境界に合わせるため、VUの末尾を0で埋める。

【００７２】

尚、本実施例では、図１３および図１４に示すAVストリーム構成を用いて説明しているが、本発明はこのストリーム構成に限定されるものではない。

【００７３】

＜AVストリーム管理方法＞

AVストリームの管理方法は、前述のQuickTimeファイルフォーマットをベースにしている。図１５にQuickTimeによるAVストリーム管理形態を示す。AAUをオーディオデータのサンプル、Sequence headerと整数個のGOPをビデオデータのサンプルとし、VU中のメインオーディオとビデオの塊をそれぞれ1チャンクに対応させる。

【００７４】

＜ディスク配置決定方法＞

AVストリームのディスク上での配置決定方法について説明する。シームレス再生を保証するためには、次へのRUへのジャンプ時間も含めたRU読み出し時間がRUの再生時間より小さければよい。

【００７５】

つまり、RU再生時間をＴe(i)、AVストリーム中の任意のRUであるRU#iについて最大再生時間をT(i)、分断ジャンプを含めた最大読み出し時間をTr(i)としたとき、Te(i)≧Tr(i)・・・＜式1＞、を満たせばよい。

【００７６】

AVストリーム中のメインオーディオ、ビデオの最大ビットレートをそれぞれRa、Rv、再生機器の最大アクセス時間をTa、連続読み出しレートをRsとしたとき、Tr(i)=Te(i)×(Rv+Ra)/Rs+Ta・・・＜式２＞、となる。

【００７７】
＜式１＞＜式２＞をTe(i)で解いて、Te(i)≧Ta×Rs/(Rs-Rv-Ra)・・・＜式３＞、が得られる。
【００７８】
つまり、シームレス再生保証可能なRU再生時間下限値Teminは、Temin=Ta×Rs/(Rs-Rv-Ra)・・・＜式４＞、となる。
【００７９】
＜記録時の処理＞
ユーザから録画が指示された場合の処理を、図１６に沿って説明する。このとき記録するAVストリームは、ビデオのビットレートRv=5Mbps、オーディオのサンプリング周波数48kHz、ビットレートRa=Rp=256kbpsであるものとする。すでに、ファイルシステムの管理情報はRAM上に読み込まれているとする。
【００８０】
まず、ストリームの構成や連続領域の構成を決定する(ステップ701)。1VUを1GOP＝30フレームで構成するとしたとき、＜式４＞にRs=20Mbps、Ta=1秒、Rv=5Mbps、Ra=256kbpsを代入し、Te(i)の範囲である1.36秒以上が得られる。1VUの再生時間を0.5秒としているため、RU再生時間は2秒とする。
【００８１】
次に、2個のVUを連続的に記録可能な空き領域を探す。具体的には2×(Rv+Ra)、つまり11 Mbit以上の連続的な空き領域を、RAM102上のSpace Bitmapを参照して探す。存在しなければ録画を中止し、録画できないことをユーザに知らせる(ステップ702)。
【００８２】
また、オーディオエンコーダ117、ビデオエンコーダ118をそれぞれ起動する(ステップ703)。次に、記録用バッファに1ECCブロック分(32KB)以上のデータが蓄積されているかどうかをチェックし(ステップ704)、蓄積されている間、ステップ705からステップ708を繰り返す。
【００８３】
蓄積されていれば、次に記録するディスク上のECCブロックの空き状況を、RAM上のSpace Bitmapを参照して調べる(ステップ705)。空きがなければ、2個のVUを記録可能な連続的な空き領域を探し(ステップ707)、その空き領域の先頭へピックアップを移動する(ステップ708)。
【００８４】
次に、記録用バッファ111中の1ECCブロック分のデータを、ディスクに記録する(ステップ706)。記録用バッファ111にデータが蓄積されていなければ、記録終了が指示されているかどうかをチェックし(ステップ709)、記録終了でなければ、ステップ704を実行する。
【００８５】
記録終了が指示されていた場合、以下のステップを実行する。まず、記録用バッファ中の32KBに満たないデータに関して、末尾にダミーデータを付加し32KBにする(ステップ710)。次に、そのデータをディスク上に記録する(ステップ711～ステップ714)。最後に、RAM102上のQuickTime管理情報（Movie atom）とファイルシステム管理情報を、光ディスク106に記録する(ステップ715～ステップ716)。
【００８６】
以上の処理と並行するオーディオエンコーダ117、ビデオエンコーダ118やマルチプレクサ113の動作について説明する。それぞれのエンコーダはマルチプレクサ113にエンコード結果を送り、マルチプレクサはそれらを多重化用バッファ114に格納する。
【００８７】
1VU分のデータ、つまり1GOPとそれに同期して再生されるAAUが多重化用バッファ114に蓄積されたら、マルチプレクサ113は記録用バッファ111に1VUのデータを送る。さらに、ホストCPU101に1VU分のデータがエンコードできたことを通知し、ホストCPU101はVUを構成するGOPやAAUの数およびサイズを基に、RAM102上のQuickTime管理情報を更新する。

【００８８】
＜編集時の処理＞
AVストリームの途中をRU単位で削除することを考える。前述のように、RUの境界はECCブロック境界と一致する。さらに、ECCブロックは16セクタで構成され、セクタと論理ブロックは一致する。したがって、ファイルシステム管理情報とQuickTime管理情報を書き換えるのみで、RU単位の削除が可能である。
【００８９】
ファイルシステム管理情報に関しては、前述のSpace bitmap中の削除する領域に対応するビットを0にすることで、領域を開放し、削除対象RUを管理するエクステントを変更する。QuickTime管理情報に関しては、削除区間に含まれるサンプルをSample tableから削除し、削除区間の後に位置するチャンクのChunk offsetの値を削除区間のバイト数マイナスする。
【００９０】
また、削除による各トラックの再生時間の短縮に伴い、Edit list（図１０，１１）のTrack durationを短縮する（図１０，１１）。上記の削除により、AVストリームは削除区間の直前と直後とが接続された形となる。このとき、ビデオトラックについては、繋ぎ目を示すために、削除直前と削除直後の領域の繋ぎ目にあたる時刻の前後で別のエントリとする。
【００９１】
例えばAVストリーム#1とAVストリーム#2を繋ぐことによって、図１７に示すような構成となったAVストリームがあった場合、Edit list atom（図１０）を、図１８に示すように、2エントリで構成する。一方、AVストリーム同士をRU単位で結合する場合も、削除の場合と同様、ファイルシステム管理情報とQuickTime管理情報とを書き換えることにより対応可能である。この情報により、トラックの時間軸における繋ぎ目の時刻および、その時刻に対応するデータのファイル中でのアドレスがわかる。
【００９２】
尚、本実施例では、繋ぎ目が1個の場合について説明したが、2個以上の場合、個数に応じてエントリ数を増やせばよいことは言うまでもない。
【００９３】
尚、ここでは、1トラックのビデオにおいて、Edit listのエントリを切り替えることで繋ぎ目を示したが、AVストリーム同士を結合することによって生じた、復号・再生に支障を来たす可能性のある箇所を示すことができればどのような方法を用いてもよいことは言うまでもない。一例を挙げると、繋ぎ目毎にビデオトラックを切り替えることで、繋ぎ目を示すことが考えられる。
【００９４】
具体的には、結合によって、図１７に示すような構成となったAVストリームがあった場合、前半部と後半部とを別トラックで管理し、前半部・後半部を管理するトラックのEdit list atom（図１０）の内容を、それぞれ図１９(a)、(b)に示す内容とすることで、繋ぎ目を示すことが可能である。このとき、後の編集時にビデオトラック＃１と＃２が誤まって統合されることを防ぐために、ビデオトラック#1、＃２の少なくとも1個の値、具体的にはCreation time（図５）等を異なる値にする。
【００９５】
また、繋ぎ目の前後で異なる内容のSample description atom（図９）を用いることで、繋ぎ目を示してもよい。具体的には、結合によって、図１７に示すような構成となったAVストリームがあった場合、図２０に示すように、AVストリーム＃1中のビデオデータとAVストリーム＃2中のビデオデータがSample description atomの異なるエントリ＃1、＃２で表される属性を参照するようにする。
【００９６】
このことにより、繋ぎ目を示すことが可能になる。さらに、Sample description atomのそれぞれのエントリ中の少なくとも1個の値が異なるようにすることで、後の編集時にS

ample description tableの最適化(内容の共通な複数のエントリを1個にまとめること)のためにエントリ＃１、＃２が誤まってマージされることを防ぐことが可能になる。

【００９７】

＜再生時の処理＞

ユーザから再生が指示された場合の処理を、図２１に沿って説明する。ここでは、すでに再生の対象となるAVストリームに関するQuickTime管理情報がRAM102に読み込まれているものとする。

【００９８】

光ディスク106上の再生指示されたVUの先頭から再生用データの読み出しを行う(ステップ901)。このとき、十分な再生時間分のデータを再生用バッファ110へ読み出すまでステップ901を繰り返す(ステップ902)。

【００９９】

ここで十分とは、再生用データ読み出しの中断期間が最大の場合でも、再生が途切れないだけのデータ量を意味する。具体的には、AVデータの読み出しに伴う分断のジャンプ(最大1秒)を行った場合を想定し、1秒分のデータ量とする。

【０１００】

次に、ホストCPU101は、ビデオデコーダ116およびオーディオデコーダ115を起動する(ステップS903)。ホストCPU101は、QuickTime管理情報に基づき、デマルチプレクサ112に指令を出すことで、符号化データを再生用バッファ110からオーディオデコーダ115およびビデオデコーダ116へ転送する。

【０１０１】

また、ユーザから再生終了を指示されていないかチェックする(ステップ904)。指示されていなければ、再生用AVデータの読み出しを行う(ステップ905)。再生終了を指示されていれば、終了する。

【０１０２】

ホストCPU101はシステムクロック105を参照し、現在の時刻がビデオトラックのEdit list atom（図１０）のエントリ（図１１）の切り替えのタイミングであった場合、その箇所が繋ぎ目であると判断し、ビデオデコーダ116およびオーディオデコーダ115の復号を一時停止する。このとき、繋ぎ目を示すのに、ビデオチャンクの参照するSample description table（図２０）のエントリの切り替わりを利用している場合は、エントリの切り替わりが発生した箇所を繋ぎ目と判断する。

【０１０３】

このことによって、繋ぎ目直前のビデオフレームが複数フレーム連続して表示(フリーズ)されることになるが、その間にビデオデコーダ116のデコーダバッファにデータを貯えることができるため、バッファのアンダーフローは起きない。つまり、デコーダバッファのアンダーフローが発生したときのように、再生がしばらく乱れるようなことはない。また、もともと繋ぎ目における映像は不連続であるため、それ以外の箇所でフリーズが発生する場合に比べ、乱れが目立たない。

【０１０４】

尚、本実施例では、ビデオデコーダが1個だけを用いているが、ビデオデコーダを複数個用意して繋ぎ目で切り替えてもよい。具体的には、再生時に以下のことを行う。まず、ビデオトラックのEdit list atom のエントリの切り替えからAVデータ中の実際の繋ぎ目の位置がわかるため、繋ぎ目以降のビデオデータを繋ぎ目以前のビデオデコーダとは別のビデオデコーダに送りデコードを開始し、いつでも表示可能な状態にしておく。次に、Edit list atomのエントリの切り替えのタイミングになった時点で、ビデオデコーダを繋ぎ目以前のものから繋ぎ目以降のものに切り替える。この場合、繋ぎ目の前後でデコーダが異なるため、VBVバッファの占有量の不連続が起こることはなく、さらに、ビデオデコーダが1個のときのようにフリーズさせる必要がない。

【０１０５】

また、本実施例では、編集の際に生じる繋ぎ目を管理しているが、本発明の管理対象で

ある繋ぎ目は編集の際に生じるものに限定されない。例えば、録画の際、新規のAVストリームを既存のAVストリームファイルの末尾に追加するかたちで行う場合、追加点の前後でVBVバッファ占有量の不連続が生じ、そのAVストリームファイルをそのまま再生した場合、追加点の直後で復号・再生が乱れるおそれがあるが、追加点を本実施例と同様に管理することで、そのような復号・再生の乱れを抑えることが可能である。

【０１０６】

以上説明したように、本発明によれば、複数のAVストリームを繋ぎ合わせて1本のAVストリームを作成した場合に、そのAVストリームを管理する管理情報によって、繋ぎ目の位置情報を管理することで、再生時に繋ぎ目付近で表示が乱れることを防ぐことが可能になる。

【産業上の利用可能性】

【０１０７】

映像データや音声データ等のAVストリームを、例えばハードディスクや光ディスク等のランダムアクセス可能な記録媒体に記録、復号する場合に、該AVストリーム間の再生データの乱れを防止できるデータ記録、編集、復号方法、並びにそれらの装置に適している。

【図面の簡単な説明】

【０１０８】

【図１】本発明の実施形態におけるビデオディスクレコーダの概略構成を示すブロック図である。

【図２】QuickTimeファイルフォーマットにおける管理情報とAVストリームの関係を示す説明図である。

【図３】QuickTimeファイルフォーマットにおけるMovie atomの概要を示す説明図である。

【図４】QuickTimeファイルフォーマットにおけるTrack atomの概要を示す説明図である。

【図５】QuickTimeファイルフォーマットにおけるTrack header atomの概要を示す説明図である。

【図６】QuickTimeファイルフォーマットにおけるMedia atomの概要を示す説明図である。

【図７】QuickTimeファイルフォーマットにおけるMedia information atomの概要を示す説明図である。

【図８】Sample table atomによるデータ管理の例を示す説明図である。

【図９】QuickTimeファイルフォーマットにおけるSample table atomの概要を示す説明図である。

【図１０】QuickTimeファイルフォーマットにおけるEdit atomの概要を示す説明図である。

【図１１】Edit atomによる再生範囲指定の例を示す説明図である。

【図１２】QuickTimeファイルフォーマットにおけるUser-defined data atomの概要を示す説明図である。

【図１３】本発明の第１の実施例におけるストリームの構成を示す説明図である。

【図１４】本発明の第１の実施例におけるVUの構造を示す説明図である。

【図１５】本発明の第１の実施例におけるQuickTimeによるAVストリーム管理形態を示す説明図である。

【図１６】本発明の第１の実施例における記録動作を示すフローチャートである。

【図１７】本発明の第１の実施例におけるAVストリーム結合の状態を示す説明図である。

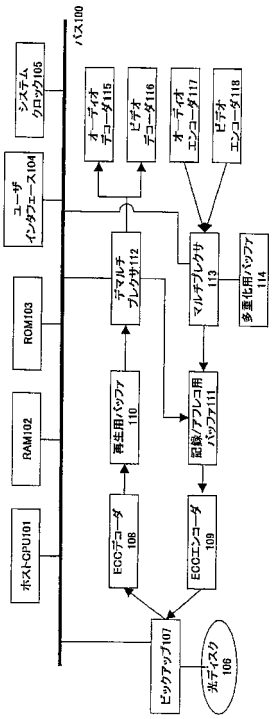
【図１８】本発明の第１の実施例におけるAVストリーム結合を管理する情報の構成の第1の例を示す説明図である。

【図１９】本発明の第１の実施例におけるAVストリーム結合を管理する情報の構成の第２の例を示す説明図である。

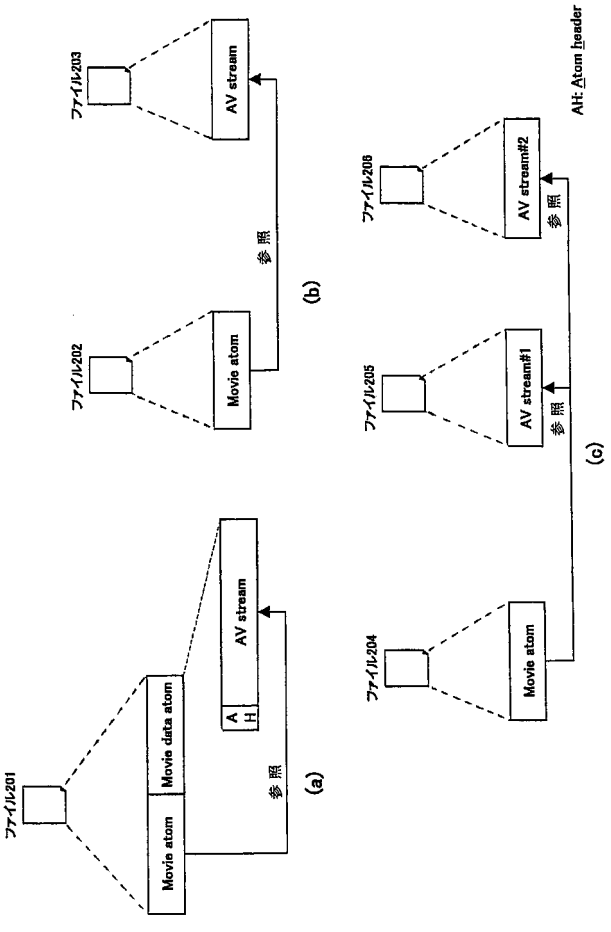
【図２０】本発明の第１の実施例におけるAVストリーム結合を管理する情報の構成の第３

の例を示す説明図である。
【図２１】本発明の第１の実施例における再生動作を示すフローチャートである。
【図２２】VBVバッファの占有量遷移の例を示す説明図である。
【図２３】従来技術によるMPEGビデオストリーム結合の例を示す説明図である。
【符号の説明】
【０１０９】
100　バス、
101　ホストCPU、
106　光ディスク、
108　ECCデコーダ、
109　ECCエンコーダ、
112　デマルチプレクサ
113　マルチプレクサ
115　オーディオデコーダ
116　ビデオデコーダ、
117　オーディオエンコーダ、
118　ビデオエンコーダ

【図１】



【図２】



【図３】

```
Movie atom {
        Atom size
        Type(='moov')
        movie header atom
        track atom (video track)
        track atom (main audio track)
                :
        User-defined data atom
}
```

【図４】

```
Track atom {
        Atom size
        Type(='trak')
        Track header  atom
        Edit atom
        Track reference atom
        Media atom
        User-defined data atom
                :
}
```

【図５】

```
Track header atom {
        Atom size
        Type(='tkhd')
        Version
        Flags
        Creation time
        Modification time
        Track ID
        Reserved
        Duration
        Reserved
        Layer
        Alternate group
        Volume
        Reserved
        Matrix structure
        Track width
        Track height
}
```

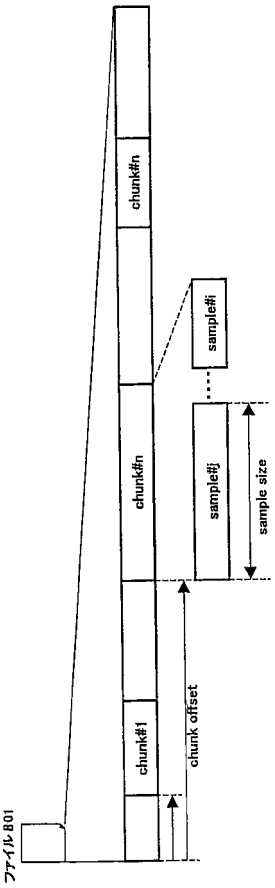
【図６】

```
Media atom {
        Atom size
        Type(='mdia')
        Media header  atom
        Handler reference atom
        Media information atom
        User-defined data atom
                :
}
```

【図７】

```
Media information atom {
        Atom size
        Type(='minf')
        {Video or Sound or Base} media information header atom
        Handler reference atom
        Data information atom
        Sample table atom
}
```

【図８】



【図９】

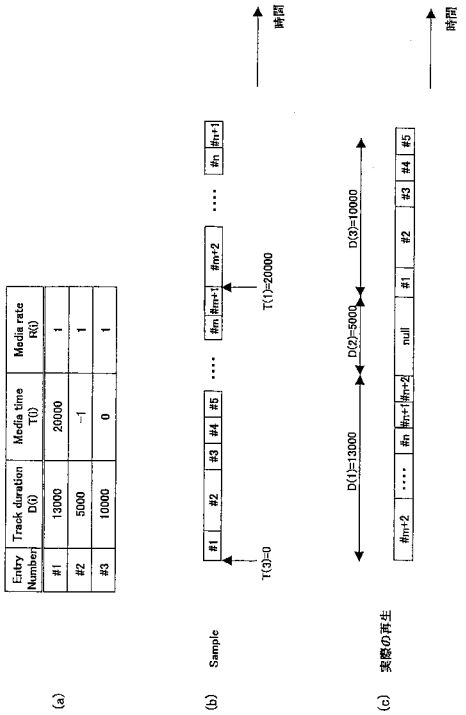
```
Sample table atom {
        Atom size
        Type(='stbl')
        Sample description atom
        Time-to-sample atom
        Sync sample atom
        Sample-to-chunk atom
        Sample size atom
        Chunk offset atom
}
```

【図１０】

```
Edit atom {
        Atom size
        Type(='edts')
        Edit list atom
}

Edit list atom {
        Atom size
        Type(='elst')
        Versions
        Flags
        Number of entries(=N)
        for (i = 0; i < N; i++){
                Track duration
                Media time
                Media rate
        }
}
```

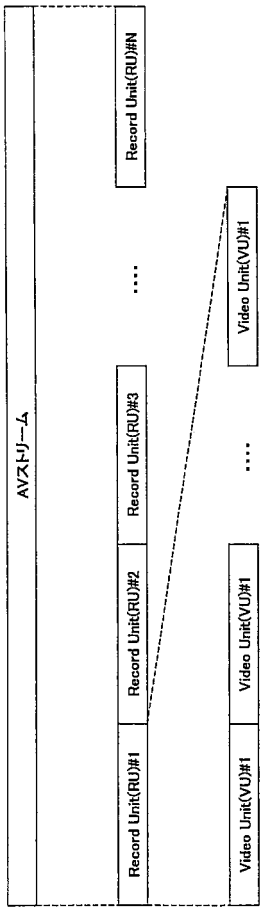
【図１１】



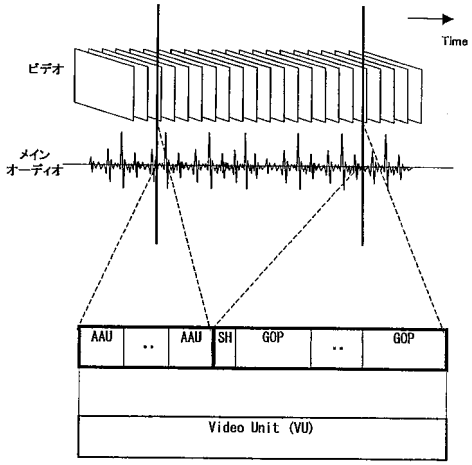
【図１２】

```
User-defined data atom {
        Atom size
        Type(='udta')
        for (i=0;i<N; i++){
                Atom size
                Type
                User data
        }
}
```

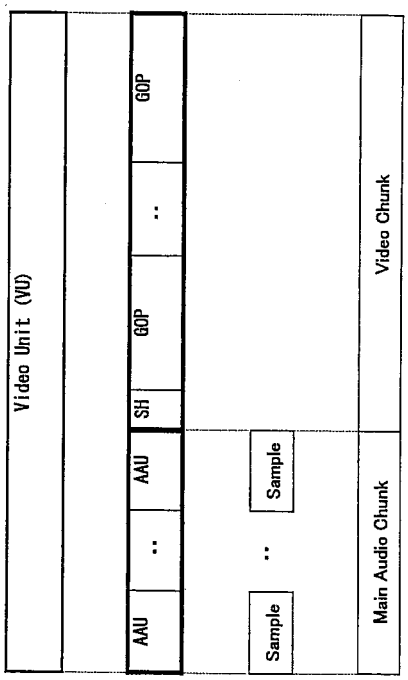
【図１３】



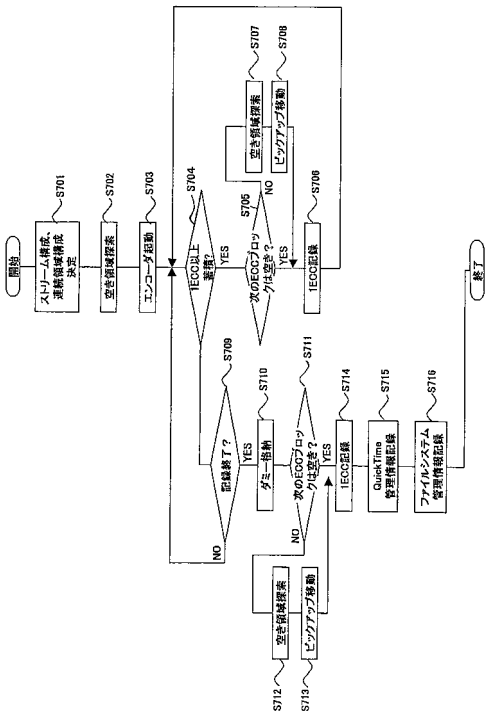
【図１４】



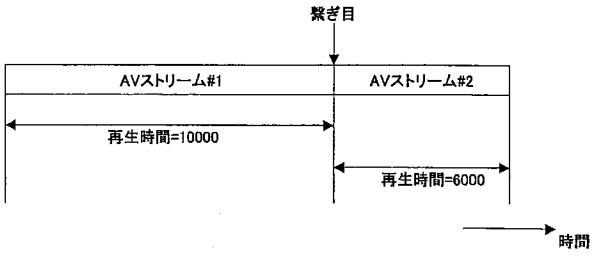
【図１５】



【図１６】



【図１７】



【図１８】

Entry Number	Track duration D(i)	Media time T(i)	Media rate R(i)
#1	10000	0	1
#2	6000	10000	1

【図１９】

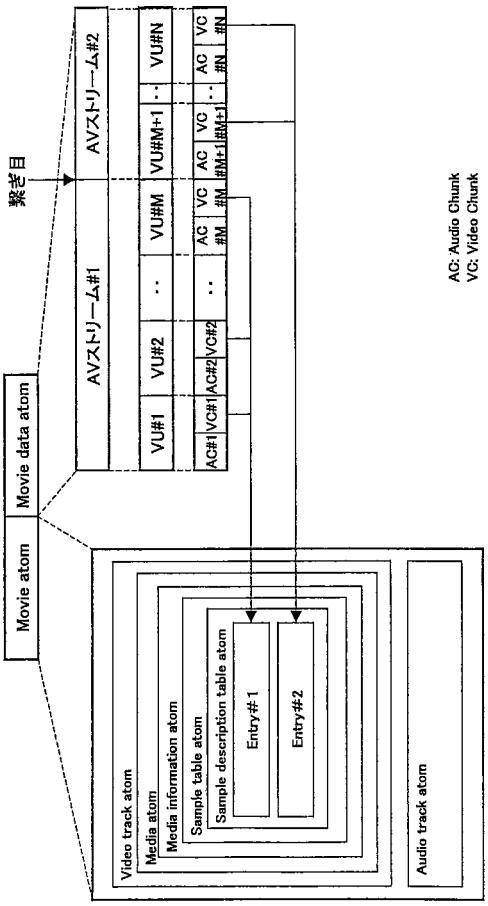
Entry Number	Track duration D(i)	Media time T(i)	Media rate R(i)
#1	10000	0	1
#2	6000	-1	1

(a) ビデオトラック#1

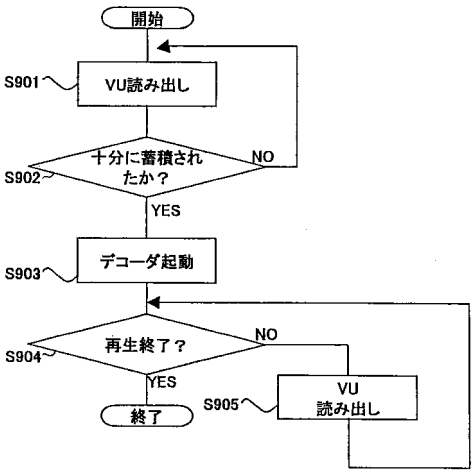
Entry Number	Track duration D(i)	Media time T(i)	Media rate R(i)
#1	10000	-1	1
#2	6000	10000	1

(b) ビデオトラック#2

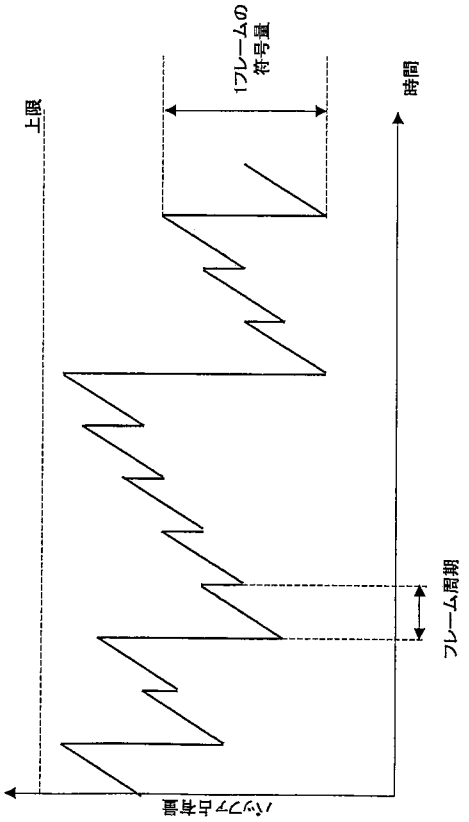
【図２０】



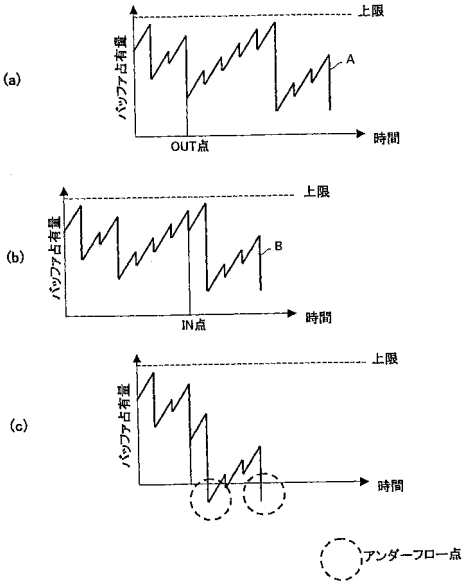
【図２１】



【図２２】


1フレームの
符号量
時間
上限
フレーム周期
バッファ占有量

【図２３】


(a)
上限
バッファ占有量
A
時間
OUT点
(b)
上限
バッファ占有量
B
時間
IN点
(c)
上限
バッファ占有量
時間
アンダーフロー点

フロントページの続き

(72)発明者　山口　孝好
大阪府大阪市阿倍野区長池町２２番２２号　シャープ株式会社内

Ｆターム(参考) 5C053 FA14 FA24 GA11 GB11 GB38
5C059 KK39 MA00 RC00 TA00 TC43 UA05
5D044 AB05 AB07 BC01 BC02 BC04 CC04 DE25 EF03 EF05 FG10
GK08 GK12 HL02 HL16
5D110 AA12 AA17 AA27 AA29 CA16 CD02 CK02 CK28