

、前記第1のAVデータ及び第2のAVデータのそれぞれを再生する際のエントリポイントとなり得るビデオユニットのそれぞれの開始時刻を示すタイムスタンプとを含み、
前記ファイルシステム管理情報を前記記録媒体から取得する第1のステップと、
前記プログラム情報ファイルを前記記録媒体から取得する第2のステップとを含むこと
 を特徴とするAVデータ再生方法。

【請求項2】

少なくとも、同一の時間軸上で切り替えて再生する1つのシーンを構成する第1のAVデータと第2のAVデータとが記録された記録媒体を再生するAVデータ再生装置であって、

前記第1のAVデータ及び第2のAVデータは、それぞれが所定の分割単位で第1の部分AVデータ及び第2の部分AVデータに分割された状態で、かつ互いに切り替えて再生する、前記第1の部分AVデータと前記第2の部分AVデータとが交互に配置するように前記記録媒体に記録されており、

さらに、前記記録媒体には、前記第1のAVデータと前記第2のAVデータとを個別のファイルとして扱うための情報を管理するファイルシステム管理情報と、前記第1のAVデータと前記第2のAVデータとを関連づけて1個のコンテンツとして扱うためのプログラム情報ファイルとが記録されており、

前記ファイルシステム管理情報は、前記第1の部分AVデータの位置情報が当該第1の部分AVデータの再生順に記録され、また前記第2の部分AVデータの位置情報が当該第2の部分AVデータの再生順に記録されたものであり、

前記プログラム情報ファイルは、前記第1のAVデータ及び第2のAVデータのファイル名を有すると共に、前記第1のAVデータ及び第2のAVデータを再生するための時刻情報を含み、

前記時刻情報は、前記シーンの開始時刻を示すタイムスタンプと、前記シーンを構成し、前記第1のAVデータ及び第2のAVデータのそれぞれを再生する際のエントリポイントとなり得るビデオユニットのそれぞれの開始時刻を示すタイムスタンプとを含み、

前記ファイルシステム管理情報を前記記録媒体から取得する手段と、

前記プログラム情報ファイルを前記記録媒体から取得する手段とを備えていることを特徴とするAVデータ再生装置。

【請求項3】

コンピュータに、少なくとも、同一の時間軸上で切り替えて再生する1つのシーンを構成する第1のAVデータと第2のAVデータとが記録された記録媒体を再生させるプログラムであって、

前記第1のAVデータ及び第2のAVデータは、それぞれが所定の分割単位で第1の部分AVデータ及び第2の部分AVデータに分割された状態で、かつ互いに切り替えて再生する、前記第1の部分AVデータと前記第2の部分AVデータとが交互に配置するように前記記録媒体に記録されており、

さらに、前記記録媒体には、前記第1のAVデータと前記第2のAVデータとを個別のファイルとして扱うための情報を管理するファイルシステム管理情報と、前記第1のAVデータと前記第2のAVデータとを関連づけて1個のコンテンツとして扱うためのプログラム情報ファイルとが記録されており、

前記ファイルシステム管理情報は、前記第1の部分AVデータの位置情報が当該第1の部分AVデータの再生順に記録され、また前記第2の部分AVデータの位置情報が当該第2の部分AVデータの再生順に記録されたものであり、

前記プログラム情報ファイルは、前記第1のAVデータ及び第2のAVデータのファイル名を有すると共に、前記第1のAVデータ及び第2のAVデータを再生するための時刻情報を含み、

前記時刻情報は、前記シーンの開始時刻を示すタイムスタンプと、前記シーンを構成し、前記第1のAVデータ及び第2のAVデータのそれぞれを再生する際のエントリポイントとなり得るビデオユニットのそれぞれの開始時刻を示すタイムスタンプとを含み、

10

20

30

40

50

前記コンピュータに、
前記ファイルシステム管理情報を前記記録媒体から取得する第1のステップと、
前記プログラム情報ファイルを前記記録媒体から取得する第2のステップとを行わせる
ことを特徴とするプログラム。

【請求項4】

請求項3に記載のプログラムを格納していることを特徴とする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、ハードディスク、光ディスク、半導体メモリ等のランダムアクセス可能な記録媒体に対して、映像データ、音声データを記録する記録方法、記録装置、並びに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のデジタル記録再生装置（以下、ビデオディスクレコーダと呼ぶ）が普及しつつある。それらにおいて、テープメディアと同様にアフターレコーディング（アフレコ）機能を安価に実現する技術が求められている。アフレコ機能は、既に記録したオーディオやビデオに対し、後から情報、特にオーディオを追記する機能である。

【0003】

ディスクメディアを用いてアフレコ機能を実現している従来技術として、当出願人による特許文献1がある。以下、図20を用いてその概要を説明する。

【0004】

特許文献1では、ストリームファイル3000は独自のストリームフォーマットであり、所定の再生時間ごとに分割したオリジナルストリームデータ（初期録画したビデオ・オーディオデータ）の間に同期再生するアフレコデータ用領域を挿入して構成する。図20（a）を例にとると、部分オリジナルストリームデータ3021の直前には、同期再生するアフレコオーディオデータを格納するためのアフレコデータ用領域3011を挿入する。3012と3022および3013と3023の間の関係も同様である。

【0005】

ストリームファイル3000を光ディスク3001に記録する際、図20（b）に示すように、同期再生する部分オリジナルストリームデータとアフレコデータ用領域とを物理的な近傍に配置することで、アフレコデータを含めた再生時にシークを最小限にすることができ、シークによる再生の途切れを抑えている。また、部分オリジナルストリームデータの再生時間を、シーク時間等を考慮しリアルタイムアフレコが可能な値に設定（数秒程度）することで、リアルタイムアフレコを保証している。

【0006】

また、世の中で広く用いられているデータ記録方法として、上記特許文献1に記載のストリーム構成とは異なる構成を持ったISO/IEC 13818-1に定義されるTransport Stream（以下MPEG-2 TS）や、Program Stream（以下MPEG-2 PS）がある。例えば、DVD-VideoはMPEG-2 PSであり、デジタル放送やIEEE-1394による機器間のデータ転送形式はMPEG-2 TSである。MPEG-2 PS/TSを考慮したアフレコに関する従来技術については、特許文献2や特許文献3がある。

【特許文献1】特開2001-43616号公報

【特許文献2】特開2000-306327号公報

【特許文献3】特開平11-298845号公報

【発明の開示】

【発明が解決しようとする課題】

【0007】

10

20

30

40

50

しかしながら、MPEG-2 PS/TSに対して、上記特許文献1に記載のストリーム構成を適用した場合、一般的に用いられているデコーダで再生しようとしても、デコードが正しくできない可能性がある。その理由を以下に説明する。

【0008】

MPEG-2 TS/PSにおいては、基準となるデコーダモデルを設定し、そのデコーダモデル中のオーディオ・ビデオのデコーダのバッファメモリがアンダーフローしたりオーバーフローしたりしないように、ビデオデータとオーディオデータとを多重化することが定められている。しかし、上記特許文献1のストリーム構成の場合、各アフレコデータ用領域には1秒以上のオーディオデータを格納することになる。このストリームファイルを一般的なMPEG-2 TS/PSデコーダで再生した場合、1秒分以上のオーディオデータがまとめて送られることになり、オーディオデコーダのバッファメモリがオーバーフローすることになる。

10

【0009】

また、特許文献2に記載のアフレコ機能は、前述のMPEG-2 PS多重化規定に従ってアフレコデータ用領域をストリーム中に多重化するというものであるが、ディスクの転送レートが低い場合にはリアルタイムアフレコが困難であるという課題がある。

【0010】

他方、特許文献3は、アフレコデータを別ファイルに記録することで、個々のファイルではMPEG-2 PS多重化規定を満たすというものである。この場合、アフレコ結果を再生する場合にアフレコデータを記録するファイルとオリジナルストリームファイルを交互に読み出すためシークを繰り返す必要がある。そのため、アフレコした結果に対し非破壊編集（ディスク上のストリームデータは動かさず、再生経路情報によって見た目上の編集を行うこと）を行った場合に、シークにより特にシーン間において再生が途切れる可能性が高まる。また、消費電力の面でも不利である。

20

【0011】

本発明は、上記課題を鑑みてなされたものであり、一般的なMPEG-2 PS/TSデコーダ上での再生やリアルタイムアフレコが可能であり、かつ、アフレコ結果を非破壊編集した場合の再生の途切れが少ないデータ記録方法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】

【0012】

本発明のAVデータ記録方法は、上記の課題を解決するために、記録媒体上に、複数のストリームデータが所定の多重化規則に従って多重化されたAVデータを、前記AVデータと同期する関連データと共に記録するAVデータ記録方法であって、前記AVデータおよび前記関連データを所定の間隔毎に部分AVデータおよび部分関連データに分割する第1のステップと、前記記録媒体上において、一続きの前記部分AVデータと前記部分関連データとを記録するための連続領域である第1の連続領域を確保する第2のステップと、前記第1の連続領域に前記部分AVデータと前記部分関連データとを連続的に記録する第3のステップと、前記部分AVデータおよび前記部分関連データとをそれぞれ別のファイルとして管理すると共に、前記部分AVデータおよび前記部分関連データとを別のファイルとして扱うための情報を管理するファイルシステム管理情報を前記記録媒体上に記録する第4のステップとを備えることを特徴としている。

30

40

【0013】

上記の構成によれば、上記記録媒体上に記録されるAVデータ（例えば、オリジナルストリーム）および関連データ（例えばアフレコデータ）は、第1のステップによって部分AVデータおよび部分関連データに分割され、シームレス再生やリアルタイムアフレコが保証される記録単位に設定される。

【0014】

分割された部分AVデータおよび部分関連データは同期するデータ同士を一続きとして、第2、第3のステップの処理によって上記記録媒体上で互いに物理的に近傍した位置に記録される。

【0015】

50

さらに、第4のステップにおいて記録されるファイルシステム管理情報では、部分AVデータおよび部分関連データが別のファイルとして管理されることによって、リアルタイムアフレコを保証し非破壊編集性能が高く、なおかつ一般的なMPEG-2 PSのデコーダでの再生が可能になる。また、AVデータの再生と関連データの再生とを同期させて行った場合に、部分AVデータおよび部分関連データが近傍して記録されていることから、シークの頻度が低くなり、その他のデータとさらに同期再生させる余地が大きい。例えば、アフレコオーディオ以外にさらにグラフィックスデータ等を非破壊編集で付加しても再生が途切れる可能性が低い。

【0016】

また、別ファイルであるファイルシステム管理情報において、記録媒体上の近傍に配置されている部分AVデータおよび部分関連データの対応関係を示す情報を記録媒体に記録することによって、ファイルシステム管理情報を参照することなく容易に連続記録されている部分AVデータおよび部分関連データを知り、最適なデータ読み込みが可能になる。

10

【0017】

また、本発明のAVデータ記録方法において、前記部分AVデータの再生開始時刻と、前記第1の連続領域の位置情報の対応情報とを前記記録媒体に記録する第5のステップを備える構成としてもよい。

【0018】

上記の構成よれば、アフレコの対象となる部分AVデータに対応する部分関連データ（アフレコ用領域）の位置を容易に特定することが可能となる。

20

【0019】

また、本発明のAVデータ記録方法において、前記部分関連データが、対応する前記部分AVデータの近傍に記録されているか否かを管理する情報を前記記録媒体に記録する第6のステップを備える構成としてもよい。

【0020】

上記部分AVデータおよび部分関連データの記録時にディフェクトに遭遇した場合、記録中の部分関連データは破棄し、別の領域に新規にCAを記録することがある。

【0021】

上記の構成によれば、その際に、その部分関連データを管理する情報において対応する部分AVデータの近傍に部分関連データが存在しないことがわかるようにしておくことで、非破壊編集時や非破壊編集結果の再生時に、部分AVデータおよび部分関連データが連続して記録されていない区間が容易に判断可能であり、その区間は再生が途切れる可能性が高いことを事前にユーザに伝えることが可能となる。

30

【0022】

また、発明の他のAVデータ記録方法は、記録媒体上に、複数のストリームデータが所定の多重化規則に従って多重化されたAVデータを、前記AVデータと同期する関連データと共に記録可能とするAVデータ記録方法であって、前記AVデータを所定の間隔毎に部分AVデータに分割する第7のステップと、前記記録媒体上において、一続きの前記部分AVデータと、前記関連データの記録時に前記部分AVデータに対応して分割される部分関連データの記録領域を確保する部分予約データとを記録するための連続領域である第1の連続領域を確保する第8のステップと、前記部分予約データを作成しながら、前記第1の連続領域に前記部分AVデータと前記部分予約データとを連続的に記録する第9のステップと、前記部分AVデータおよび前記部分予約データとをそれぞれ別のファイルとして管理すると共に、前記部分AVデータおよび前記部分予約データを別のファイルとして扱うための情報を管理するファイルシステム管理情報を前記記録媒体上に記録する第10のステップとを備えることを特徴としている。

40

【0023】

上記の構成によれば、上記記録媒体上に記録されるAVデータ（例えば、オリジナルストリーム）は、第7のステップによって部分AVデータに分割され、シームレス再生やリアルタイムアフレコが保証される記録単位に設定される。

50

【 0 0 2 4 】

分割された部分 A V データは、これと同期する部分関連データの記録領域を確保するための部分予約データと一続きとして、第 8 , 第 9 のステップの処理によって上記記録媒体上で互いに物理的に近傍した位置に記録される。

【 0 0 2 5 】

さらに、第 1 0 のステップにおいて記録されるファイルシステム管理情報では、部分 A V データおよび部分予約データが別のファイルとして管理されることによって、関連データの記録時におけるリアルタイムアフレコを保証し非破壊編集性能の高く、なおかつ一般的な MPEG-2 PS のデコーダでの再生が可能になる。

【 0 0 2 6 】

また、上記 A V データ記録方法においては、前記関連データの記録時に、前記関連データを所定の間隔毎に部分関連データに分割する第 1 1 のステップと、前記部分関連データを、対応する部分 A V データと連続して記録されている前記部分予約データの確保領域に記録する第 1 2 のステップと、前記部分関連データを前記部分 A V データおよび前記部分予約データとは別のファイルとして管理すると共に、前記部分関連データを別のファイルとして扱うための情報を管理するファイルシステム管理情報を前記記録媒体上に記録する第 1 3 のステップとを備える構成とすることができる。

【 0 0 2 7 】

上記の構成によれば、A V データの再生と関連データの再生とを同期させて行った場合に、部分 A V データおよび部分関連データが近傍して記録されていることから、シークの頻度が低くなり、その他のデータとさらに同期再生させる余地が大きい。例えば、アフレコオーディオ以外にさらにグラフィックスデータ等を非破壊編集で付加しても再生が途切れる可能性が低い。

【 0 0 2 8 】

また、別ファイルであるファイルシステム管理情報において、記録媒体上の近傍に配置されている部分 A V データおよび部分関連データの対応関係を示す情報を記録媒体に記録することによって、ファイルシステム管理情報を参照することなく容易に連続記録されている部分 A V データおよび部分関連データを知り、最適なデータ読み込みが可能になる。

【 発明の効果 】

【 0 0 2 9 】

本発明の A V データ記録方法は、以上のように、前記 A V データおよび前記関連データを所定の間隔毎に部分 A V データおよび部分関連データに分割する第 1 のステップと、前記記録媒体上において、一続きの前記部分 A V データと前記部分関連データとを記録するための連続領域である第 1 の連続領域を確保する第 2 のステップと、前記第 1 の連続領域に前記部分 A V データと前記部分関連データとを連続的に記録する第 3 のステップと、前記部分 A V データおよび前記部分関連データとをそれぞれ別のファイルとして管理すると共に、前記部分 A V データおよび前記部分関連データを別のファイルとして扱うための情報を管理するファイルシステム管理情報を前記記録媒体上に記録する第 4 のステップとを備える構成である。

【 0 0 3 0 】

それゆえ、上記記録媒体上に記録される A V データおよび関連データは、第 1 のステップによって部分 A V データおよび部分関連データに分割され、分割された部分 A V データおよび部分関連データは同期するデータ同士を一続きとして、第 2 , 第 3 の処理によって上記記録媒体上で互いに物理的に近傍した位置に記録される。さらに、第 4 のステップにおいて記録されるファイルシステム管理情報では、部分 A V データおよび部分関連データが別のファイルとして管理される。

【 0 0 3 1 】

これにより、リアルタイムアフレコを保証し非破壊編集性能の高く、なおかつ一般的な MPEG-2 PS のデコーダでの再生が可能になると共に、A V データの再生と関連データの再生とを同期させて行った場合に、部分 A V データおよび部分関連データが近傍して記録さ

10

20

30

40

50

れていることから、シークの頻度が低くなるといった効果を奏する。

【0032】

また、発明の他のAVデータ記録方法は、以上のように、記録媒体上に、複数のストリームデータが所定の多重化規則に従って多重化されたAVデータを、前記AVデータと同期する関連データと共に記録可能とするAVデータ記録方法であって、前記AVデータを所定の間隔毎に部分AVデータに分割する第7のステップと、前記記録媒体上において、一続きの前記部分AVデータと、前記関連データの記録時に前記部分AVデータに対応して分割される部分関連データの記録領域を確保する部分予約データとを記録するための連続領域である第1の連続領域を確保する第8のステップと、前記部分予約データを作成しながら、前記第1の連続領域に前記部分AVデータと前記部分予約データとを連続的に記録する第9のステップと、前記部分AVデータおよび前記部分予約データとをそれぞれ別のファイルとして管理すると共に、前記部分AVデータおよび前記部分予約データを別のファイルとして扱うための情報を管理するファイルシステム管理情報を前記記録媒体上に記録する第10のステップとを備える構成である。

10

【0033】

また、上記AVデータ記録方法においては、前記関連データの記録時に、前記関連データを所定の間隔毎に部分関連データに分割する第11のステップと、前記部分関連データを、対応する部分AVデータと連続して記録されている前記部分予約データの確保領域に記録する第12のステップと、前記部分関連データを前記部分AVデータおよび前記部分予約データとは別のファイルとして管理すると共に、前記部分関連データを別のファイルとして扱うための情報を管理するファイルシステム管理情報を前記記録媒体上に記録する第13のステップとを備える構成とすることができる。

20

【0034】

それゆえ、上記記録媒体上に記録されるAVデータは、シームレス再生やリアルタイムアフレコが保証される記録単位の部分AVデータに分割され、分割された部分AVデータは部分予約データと一続きとして、上記記録媒体上で互いに物理的に近傍した位置に記録される。また、関連データの記録時は、上記関連データを部分関連データに分割し、それぞれの部分関連データを対応する部分予約データの確保領域に記録することによって行われる。

30

【0035】

これにより、AVデータの再生と関連データの再生とを同期させて行った場合に、部分AVデータおよび部分関連データが近傍して記録されていることから、シークの頻度が低くなるといった効果を奏する。

【発明を実施するための最良の形態】

【0036】

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。ここでの説明は、本発明において共通に用いる構成、個々の実施形態に固有の内容という順に行っていく。

【0037】

<システム構成>

40

図2は、後述する各実施形態において共通のビデオディスクレコーダの基本システム構成図である。

【0038】

このビデオディスクレコーダは、図2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(Error Correcting Coding)デコーダ108、ECCエンコーダ109、オーディオ再生用バッファ110、ビデオ再生用バッファ111、デマルチプレクサ112、マルチプレクサ113、記録用バッファ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、オーディオ記録用バッファ119、ビデオ記録用バッファ120

50

、デマルチプレクサ 1 2 1、アフレコデータ再生用バッファ 1 2 2、分割処理部 1 2 3 (AVデータと部分関連データに分割する手段)、空き領域管理部 1 2 5 (連続領域を確保する手段)、管理情報処理部 1 2 6 および図示しないカメラ、マイク、スピーカ、ディスプレイ等で構成される。また、ピックアップ 1 0 7、ECCデコーダ 1 0 8、ECCエンコーダ 1 0 9は、ドライブ 1 2 7 (部分AVデータと部分関連データとを連続的に記録する手段、ファイルシステム管理情報を前記記録媒体上に記録する手段)を構成する。

【0039】

ホストCPU 1 0 1は、デマルチプレクサ 1 1 2、マルチプレクサ 1 1 3、ピックアップ 1 0 7、オーディオデコーダ 1 1 5、ビデオデコーダ 1 1 6、オーディオエンコーダ 1 1 7、ビデオエンコーダ 1 1 8の制御をバス 1 0 0を通じて行う。

10

【0040】

再生時に、光ディスク 1 0 6からピックアップ 1 0 7を通じて読み出されたデータは、ECCデコーダ 1 0 8によって誤り訂正される。誤り訂正されたデータは、管理情報処理部 1 2 6によってファイルシステム管理情報が処理されて、デマルチプレクサ 1 1 2あるいはデマルチプレクサ 1 2 1に送られる。

【0041】

デマルチプレクサ 1 1 2はホストCPU 1 0 1からの指示に従い、読み出されたデータをオーディオデータかビデオデータかに応じてオーディオ再生用バッファ 1 1 0およびビデオ再生用バッファ 1 1 1に振り分ける。同様に、デマルチプレクサ 1 2 1はホストCPU 1 0 1からの指示に従い、読み出されたデータをアフレコデータ再生用バッファ 1 2 2

20

【0042】

オーディオデコーダ 1 1 5は、ホストCPU 1 0 1からの指示に従って、オーディオ再生用バッファ 1 1 0およびアフレコデータ再生用バッファ 1 2 2からデータを読み出し、読み出したデータに対するデコードを行う。同様に、ビデオデコーダ 1 1 6は、ホストCPU 1 0 1からの指示に従って、ビデオ再生用バッファ 1 1 1からデータを読み出し、読み出したデータに対するデコードを行う。

【0043】

一方、記録時に、オーディオエンコーダ 1 1 7およびビデオエンコーダ 1 1 8によって圧縮符号化されたデータは、それぞれオーディオ記録用バッファ 1 1 9およびビデオ記録用バッファ 1 2 0に送られる。マルチプレクサ 1 1 3は、ホストCPU 1 0 1からの指示に従って、オーディオ記録用バッファ 1 1 9およびビデオ記録用バッファ 1 2 0からデータを読み出し、読み出したデータをAV多重化し分割処理部 1 2 3に送る。分割処理部 1 2 3はAV多重化されたデータを所定の間隔毎に分割して記録用バッファ 1 1 4に送る。このとき、空き領域管理部 1 2 5はデータを記録するための連続領域を確保し、ECCエンコーダ 1 0 9は、記録用バッファ 1 1 4から読み出したAV多重化データに対して誤り訂正符号を付加し、ピックアップ 1 0 7を通じて光ディスク 1 0 6の確保された連続領域に記録する。

30

【0044】

オーディオデータの符号化方式にはISO/IEC 13818-3で規定されるMPEG-1 Layer-IIを、ビデオデータの符号化方式にはISO/IEC 13818-2で規定されるMPEG-2をそれぞれ用いる。光ディスク 1 0 6は、DVD-RAMのような書き換え可能な光ディスクであり、2048 byte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。

40

【0045】

<ファイルシステム>

本発明の説明において用いるファイルシステムのフォーマットである、UDF (Universal Disk Format) について図3を用いて説明する。図3(a)に示すディレクトリ/ファイル構成をUDFで記録した例を図3(b)に示す。

【0046】

図中のAVDP (Anchor Volume Descriptor Pointer) 602は、UDFの管理情報を

50

探すためのエントリポイントに相当し、通常 256 セクタ目、N セクタ目あるいは N - 256 セクタ目 (N は最大論理セクタ番号) に記録する。VDS (Volume Descriptor Sequence) 601 は、UDF が管理する領域であるボリュームに関する管理情報を記録する。ボリュームは一般に一枚のディスクに 1 個存在し、その中にパーティションを一般に 1 個含む。FSD (File Set Descriptor) 603 は、パーティションに 1 個存在する。パーティションの中での位置情報はパーティションの先頭からのセクタ番号に相当する論理ブロック番号で示される。なお、1 個の論理ブロックは 1 セクタに対応する。また、各パーティションには図示しないが Space Bitmap と呼ばれる各論理ブロックがファイルにすでに割り当てられているかそうでないかを示すテーブルが存在する。

【0047】

FSD 603 は、ルートディレクトリの File Entry (FE) である FE 604 の位置情報 (論理ブロック番号と論理ブロック数で構成され “extent” と呼ばれる) を含む。FE は、extent の集合を管理しており、extent を書き換えたり、追加したり、削除することで、ファイルを構成する実データの順番を変えたり、データを挿入したり削除したりすることが可能である。

【0048】

FE 604 はルートディレクトリの直下のファイルやディレクトリの名称等を格納する File Identifier Descriptor (FID) の集合を格納する領域 605 を管理する。領域 605 中の FID 611、FID 612 は、それぞれファイル 621、ファイル 622 のファイル名や extent の集合を管理する FE 606、FE 608 の位置情報を含む。FE 606 はファイル 621 の実データを構成する領域である領域 607、領域 610 を extent として管理する。このときファイル 621 の実データにアクセスするためには、AVDP 602、VDS 601、FSD 603、FE 604、FID 611、FE 606、領域 607、領域 610 の順にリンクを辿っていけばよい。

【0049】

〔実施の形態 1〕

本発明の第 1 の実施形態について、図 1 および図 4 ~ 図 18 を用いて説明する。

【0050】

<ファイル・ディレクトリ構成>

本実施の形態 1 に係るファイル・ディレクトリ構成について、図 4 を用いて説明する。本実施の形態 1 に関するデータは、図 4 に示すように 5 種類のファイルに格納される。

【0051】

オリジナルストリームファイル (SHRP0001.M2P) は 1 回の録画によって作成されるファイルであり、MPEG-2 PS (Program Stream) 形式である。アフレコデータファイル (SHRP0001.PRE) は、アフレコ用の領域を確保し、アフレコデータを格納するためのファイルである。オリジナルストリーム管理情報ファイル (SHRP0001.OMI) は、オリジナルストリームファイルに関する時間 - アドレス対応情報と属性情報およびアフレコデータファイルに関する属性情報やオリジナルストリームファイルとの対応関係情報とを格納するファイルであり、オリジナルストリームファイル 1 個につき 1 個ある。プログラム情報ファイル (SHRP0001.PGM) は、上記のストリームやデータのどの区間をどのような順序で再生するかを指定する情報を格納するファイルである。なお、プログラムは 1 個のコンテンツに相当し、ユーザが再生を指示する対象である。

【0052】

録画時には、上記 4 個のファイルを新規に作成する。これらのファイルは、ファイル間の関係を示すため、ファイル名の拡張子以外を共通にする。オーディオアフレコ時には、アフレコオーディオデータをアフレコデータファイル中の所定の位置に上書きし、アフレコデータ管理情報ファイルにおいてもそのことを反映する。さらに、追加したアフレコオーディオデータも再生対象にするようプログラム情報ファイルも変更する。

【0053】

また、非破壊編集時には、プログラム情報ファイルを新規に作成し、再生したい区間に

10

20

30

40

50

関して、再生したいデータを管理するオリジナルストリーム管理情報ファイルやアフレコデータ管理情報ファイルのファイル名および再生したい区間を順に記録する。なお、各ファイルのデータ構造については後述する。

【 0 0 5 4 】

< A V ストリームの形態 >

本実施の形態 1 において用いる A V ストリームの構成について、図 5 を用いて説明する。

【 0 0 5 5 】

まず、オリジナルストリームファイルについて図 5 を用いて説明する。オリジナルストリームファイルの内容は MPEG-2 PS 形式であり、図 5 (a) に示すように整数個の Continuous Unit (C U) で構成される。C U はディスク上で連続的に記録する単位である。C U の長さは、A V ストリームを構成する C U をどのようにディスク上に配置してもシームレス再生 (再生中に画像や音声が途切れないで再生できること) やリアルタイムアフレコ (アフレコ対象のビデオをシームレス再生しながらオーディオを記録すること) 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。

【 0 0 5 6 】

C U は、図 5 (b) に示すように整数個の Video Unit (V U) で構成される。V U は単独再生可能な単位であり、再生の際のエントリーポイントとなり得る。V U は、図 5 (c) に示すように整数個のオーディオパック (A # 1 ~ A # K)、ビデオパック (V # 1 ~ V # L) で構成され、それぞれ MPEG-2 PS 形式のデコーダモデルが破綻しないように A V 多重される。ディスク読み出しの際に余分なデータを読み出さずに済むように、パックのサイズはセクタサイズに一致させる。なお、パック化されるビデオデータは、1 ~ 2 個の GOP で構成されるようにし、オーディオデータは、整数個の A A U (Audio Access Unit) をパック化するようにする。

【 0 0 5 7 】

尚、GOP は、MPEG-2 ビデオ規格における画像圧縮の単位であり、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。A A U は MPEG-1 Layer-II 規格における音声圧縮の単位で、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 48 kHz の場合、A A U 1 個あたりの再生時間は 0.024 秒となる。また、V U 単位で独立再生を可能とするために、V U 中のビデオデータの先頭には Sequence Header (S H) を置く。

【 0 0 5 8 】

なお、C U は整数個の ECC ブロックで構成されるように、C U の末尾の V U は、パディングパッケージを格納したパックでパディングする。

【 0 0 5 9 】

< アフレコデータファイル >

アフレコデータファイルの構成について図 6 を用いて説明する。アフレコデータファイルは、図 6 に示すように整数個の Continuous Area (C A) で構成される。上述したオリジナルストリームファイルにおける C U 1 個は C A 1 個に対応して存在し、C A には対応する C U 中の再生データに関するアフレコデータを記録する。たとえば、オリジナルストリームファイル中の C U # n と同期して再生したいアフレコオーディオデータは C A # n に記録する。C A は整数個の ECC ブロックで構成する。

【 0 0 6 0 】

アフレコデータファイルは、オリジナルストリームファイルと同様 MPEG-2 PS 形式であり、初期録画時にはパディングパッケージを記録し、アフレコ後にはアフレコデータを格納したパックを上書きする。上書きするパックのパックヘッダ中の S C R (System Clock Reference) およびパッケージヘッダ中の P T S (Presentation Time Stamp) は、オリジナルストリームファイル中の対応するオーディオパックのものと合わせる。これにより、C A 中のオーディオパックで、C U 中の対応するオーディオパックを上書きすることで、容易にオリジナルストリームのオーディオをアフレコデータで差し替えることができる。

10

20

30

40

50

【 0 0 6 1 】

< ディスクへの配置 >

オリジナルストリームファイルおよびアフレコデータファイルのディスクへの配置について図 1 を用いて説明する。図 1 (a) に示す互いに対応するオリジナルストリームファイル (SHRP0001.M2P) とアフレコデータファイル (SHRP0001.PRE) とは、光ディスク 1 0 6 で C U の直前に対応する C A が配置されるように記録する (図 1 (b)) 。

【 0 0 6 2 】

これにより、同期再生を行うデータ同士 (C A および C U) がディスク上の近傍に置かれるため、再生時のピックアップの移動が最小限となり、後述するように非破壊編集結果を再生する際に再生が途切れる可能性が減少する。また、サイズの小さい C A を C U より前に読み出されるよう配置することによって、同期再生のためのバッファメモリ量を抑えることが可能である。

【 0 0 6 3 】

< C U 単位決定方法 >

C U 再生時間の決定方法について、図 7 および図 8 を用いて説明する。この決定方法では、機器間での互換性確保のため、基準となるデバイス (リファレンス・デバイス・モデル) と基準となるアフレコアルゴリズム (リファレンス・アフレコ・アルゴリズム) とを想定し、次にそれらを用いてアフレコを行った際にシームレス再生が破綻しないように C U 再生時間を決める。

【 0 0 6 4 】

まず、リファレンス・デバイス・モデルについて図 7 を用いて説明する。リファレンス・デバイス・モデルは 1 個のピックアップ (図示せず) と、それにつながる E C C エンコーダ・デコーダ 5 0 1、トラックバッファ 5 0 2、デマルチプレクサ 5 0 3、アフレコ用バッファ 5 0 4、オーディオエンコーダ 5 0 9、ビデオバッファ 5 0 5、オーディオバッファ 5 0 6、ビデオデコーダ 5 0 7、オーディオデコーダ 5 0 8 とによって構成される。

【 0 0 6 5 】

上記リファレンス・デバイス・モデルでは、ピックアップが 1 個であるため、再生用データのディスク 5 0 0 からの読み出しと、アフレコ用データのディスク 5 0 0 への記録は時分割で行う。ディスク 5 0 0 から再生用データを読み出す際、C A も含めて読み出す。読み出された C A を含む E C C ブロック (C A ブロック) は、トラックバッファ 5 0 2 からアフレコ用バッファ 5 0 4 に送られる。

【 0 0 6 6 】

オーディオエンコーダ 5 0 9 は、A A U 周期でアフレコ用バッファ 5 0 4 にアフレコ用データを出力する。この出力によって、アフレコ用バッファ 5 0 4 中の対応する C A ブロックを上書きする。アフレコ用データの記録は、C A ブロックを所定の E C C ブロックに記録することで行う。

【 0 0 6 7 】

ここで、オーディオフィームデータの E C C エンコーダ 5 0 1 へのデータの入力速度および E C C デコーダ 5 0 1 からのデータの出力速度を R_s とする。また、アクセスによる読み出し、記録の停止する最大期間を T_a とする。尚、この期間にはシーク時間、回転待ち時間、アクセス後に最初にディスクから読み出したデータが E C C デコーダ 5 0 1 から出力されるまでの時間が含まれる。本実施の形態 1 では $R_s = 20 \text{ Mbps}$ 、 $T_a = 1 \text{ 秒}$ とする。

【 0 0 6 8 】

次に、リファレンス・アフレコ・アルゴリズムについて、図 8 を用いて説明する。尚、図 8 中の (1) から (6) までの番号は、以下の説明中の (1) から (6) までの番号に対応する。アルゴリズムの概要は次の通りである。

(1) 再生用データの読み出しを行う。

(2) N 番目の C A である C A (N) に対応するオーディオデータのエンコードが終了すると同時に、C A (N) へのアクセスを行う。

10

20

30

40

50

(3) C A (N) をディスクに記録する。

(4)元の読み出し位置に戻る。

(5)再生用データの読み出しを行う。

(6) N + 1 番目の C A である C A (N + 1) に対応するオーディオデータのエンコードが終了すると同時に、 C A (N + 1) へのアクセスを行う。

以降は(3)～(6)の動作を繰り返す。

【 0 0 6 9 】

前記リファレンス・デバイス・モデルにおいて、前記リファレンス・アフレコ・アルゴリズムを用いてアフレコを行った場合、次のような条件を満たせば、アフレコ用バッファ 5 0 4 のオーバーフローおよびトラックバッファ 5 0 2 のアンダーフローがないことが保証できる。

【 0 0 7 0 】

すなわち、 A V ストリーム中の任意の C U である C U # i について最大再生時間を $T_e(i)$ 、分断ジャンプを含めた最大読み出し時間を $T_r(i)$ 、 C U # i に対応する C A である C A # i の最大記録時間を $T_w(i)$ としたとき、以下の式 (1) が成立すればよい。

【 0 0 7 1 】

【数 1】

$$T_e(i) \geq T_r(i) + T_w(i) \quad \dots(1)$$

【 0 0 7 2 】

なぜなら、上記式 (1) は、シームレス再生の十分条件である以下の式 (2) を満たすためである。また、以下の式 (2) において、 T_a は C A への往復のアクセスにかかる最大アクセス時間である。

【 0 0 7 3 】

【数 2】

$$\sum_i T_e(i) \geq \sum_i (T_r(i) + T_w(i)) \quad \dots(2)$$

【 0 0 7 4 】

また、 C A エンコード完了に同期してアフレコデータのディスクへの記録を行っているため、アフレコ用バッファ 5 0 4 中のデータが累積していくことはなく、アフレコ用バッファ 5 0 4 のオーバーフローもない。

【 0 0 7 5 】

式 (1) 中の $T_r(i)$ は、オリジナルストリームの最大ビットレートとアフレコオーディオストリームの最大ビットレートをそれぞれ R_o 、 R_a としたとき、以下の式 (3) を満たす。また、 R_s はオーディオフィールドデータの入力速度および出力速度、すなわちオーディオビットレートを示している。

【 0 0 7 6 】

【数 3】

$$T_r(i) = T_e(i) \times R_o / R_s + T_e(i) \times R_a / R_s + T_a \quad \dots(3)$$

【 0 0 7 7 】

上記式 (3) の右辺第 1 項、第 2 項はそれぞれ C U 中の V U 読み出し時間および C A 読み出し時間を表す。右辺第 3 項は読み出しに伴う分断ジャンプによるアクセス時間を表す。 C U 読み出し中の分断ジャンプは最大 1 回であるため、上記式 (3) は、すなわち $T_r(i)$ は、 1 回分のアクセス時間を示している。

【 0 0 7 8 】

また、 $T_w(i)$ は、以下の式 (4) を満たす。

【 0 0 7 9 】

10

20

30

40

【数4】

$$T_w(i) = 2T_a + T_e(i) \times R_a / R_s \quad \dots(4)$$

【0080】

ここで、上記式(4)の右辺第1項はCAへの往復アクセス時間を示す。CAへの往復のアクセス時間に最大アクセス時間 T_a を用いているのは、CA単位で任意の位置に記録可能にしているため、現在読み出し中のCUがディスクの最内周であり記録対象のCAがディスクの最外周ということも考えられ、最大値で見積もる必要があるためである。

【0081】

尚、前述のようにCAをディスク上で連続的に記録するようにしているため、CA記録中のアクセスは発生しない。このことにより、CA記録に伴う時間を短くすることができ、結果としてCU再生時間の下限値を低く抑えることが可能となる。

10

【0082】

式(1)に式(3)および式(4)を代入して $T_e(i)$ で解くと、リアルタイムアフレコを保證可能な $T_e(i)$ の条件、すなわち以下の式(5)が得られる。また、 R_v はビデオフレームデータの入力速度および出力速度、すなわちビデオビットレートを示している。

【0083】

【数5】

$$T_e(i) \geq (3T_a \times R_s) / (R_s - R_o - 2R_a) \quad \dots(5)$$

20

【0084】

つまり、アフレコ保證可能なCU再生時間下限値 T_{emin} は、以下の式(6)に示すものとなる。

【0085】

【数6】

$$T_{emin} = (3T_a \times R_s) / (R_s - R_o - 2R_a) \quad \dots(6)$$

【0086】

このとき、CU再生時間の上限値 T_{emax} は、以下の式(7)のように設定する。ここで、 T_{vmax} はVUの最大再生時間である。

30

【0087】

【数7】

$$T_{emax} = (3T_a \times R_s) / (R_s - R_o - 2R_a) + T_{vmax} \quad \dots(7)$$

【0088】

CU再生時間の上限値を設定するのは、アフレコ用音声と通常音声との同期再生に必要な遅延用メモリの最大量を見積り可能にし、再生互換性を保証するためである。尚、本実施の形態1では、オーディオビットレート R_a およびビデオビットレート R_v に応じて多重化間隔下限値 T_{emin} を設定しているが、ビットレートに関わらず一定の下限値を設定しても良い。ただし、その値は最大のビットレートに基づいたものでなければならない。

40

【0089】

また、CU再生時間が上記の制限を満たせば、ストリーム中のVU再生時間は固定でも可変でも構わない。

【0090】

また、本実施の形態1では、分断ジャンプと過去のCUへのピックアップの移動とを非同期に行うことを想定している。この理由は、非同期に行った方が同期して行った場合に比べ、リアルタイムアフレコを行うための条件として厳しい(再生用データの読み出しが途切れる期間が長い)ため、非同期でリアルタイムアフレコが可能であれば同期でも可能

50

であり、実装の自由度を高めることが可能になるためである。

【 0 0 9 1 】

従って、分断ジャンプと過去のC Uへのピックアップの移動を同期して行うことを前提にT e m i nを設定しても良い。この場合、式(3)の右辺第2項を取り除いて考えれば良い。

【 0 0 9 2 】

< 管理情報ファイルフォーマット >

本発明に係る管理情報ファイルフォーマットについて図9ないし図15を用いて説明する。

【 0 0 9 3 】

まず、オリジナルストリーム管理情報ファイルについて説明する。オリジナルストリーム管理情報ファイルは、図9に示すように、このファイルが管理するオリジナルストリームファイル全体に関する属性情報を格納するo__attribute()、V Uに関する情報を格納するvideo__unit__table()、このファイルが管理するアフレコデータファイル全体に関する属性情報を格納するp__attribute()、およびC Aに関する情報を格納するcontinuous__area__table()で構成される。

【 0 0 9 4 】

video__unit__table()は、図10(a)に示すように、V Uの数を示すnumber__of__video__unitと各V Uに関する情報を格納するvideo__unit__info()とで構成される。

【 0 0 9 5 】

video__unit__info()は、図10(b)に示すように、所定のV Uに関する各種属性情報を示すVU__flags、所定のV Uの先頭表示フレームのP T S (Presentation Time Stamp)を格納するVU__PTS、およびファイルの先頭からの相対パック番号を格納するVU__PNで構成される。VU__PTSおよびVU__PNによって特定のP T Sに対応するV Uの位置を特定することが可能になる。すなわち、VU__PTSはオリジナルストリーム(A Vデータ)の再生開始時刻を示し、VU__PNはC A及びC Uを記録する第1の連続領域の位置情報、言い換えるとC Aの先頭位置情報を示す。

【 0 0 9 6 】

VU__flags()は、図11(a)に示すようにフラグfirst__unit__flagを含む。first__unit__flagは1 b i tの情報であり、図11(b)に示すように、0 bの場合には管理するV UがC Uの先頭でないことを意味し、1 bの場合には管理するV UがC Uの先頭であることを意味する。

【 0 0 9 7 】

continuous__area__table()は、図12(a)に示すように、C Aの数を示すnumber__of__continuous__areaと各C Aに関する情報を格納するcontinuous__area__info()とで構成される。

【 0 0 9 8 】

continuous__area__info()は、図12(b)に示すように、所定のC Aに関する各種属性情報を示すCA__flags、所定のC Aに関して対応するC Uの先頭表示フレームのP T S (Presentation Time Stamp)を格納するCA__PTS、およびファイルの先頭からの相対パック番号を格納するCA__PNで構成される。CA__PTSおよびCA__PNによって、オリジナルストリーム中の特定のP T Sに対応するC Aの位置を特定することが可能になる。

【 0 0 9 9 】

CA__flags()は、図13(a)に示すようにフラグplacement__flagを含む。placement__flagは1 b i tの情報であり、図13(b)に示すように、0 bの場合には管理するC Aが対応する(同期再生する)C Uの直前にないことを意味し、1 bの場合には管理するC Aが対応する(同期再生する)C Uの直前にあることを意味する。

【 0 1 0 0 】

このフラグを参照することで、非破壊編集結果を再生する際に再生が途切れる可能性を知ることが可能である。すなわち、このフラグが0 bであれば、C Aへのシークが発生し

10

20

30

40

50

、再生が途切れる可能性が高いことがわかる。

【0101】

なお、o_attribute()およびp_attribute()については説明を省略する。

【0102】

最後にプログラム情報ファイルについて説明する。プログラム情報ファイルは図14に示すように、プログラム情報全般の属性情報を格納するpg_attribute()と、プログラムを構成する各シーンに関する情報を格納するscene_table()とで構成される。

【0103】

scene_table()は、図15(a)に示すように、シーン数を格納するnumber_of_sceneと各シーンに関する情報を格納するscene_info()とで構成される。scene_info()は図15(b)に示すように、所定のシーンを含むオリジナルストリームファイルを管理するオリジナルストリーム管理情報ファイルのファイル名を格納するsc_filename、所定のシーンをそのオリジナルストリームのどこから再生するかを格納するsc_start PTS、および所定のシーンの再生時間を格納するsc_durationで構成される。

10

【0104】

<記録時の処理>

次に、ユーザから録画が指示された場合の処理を、図16のフローチャートに沿って説明する。このとき記録するAVストリームはビットレート $R_o = 12\text{Mbps}$ 、オーディオのビットレート $R_a = 256\text{kbps}$ で、VU再生時間固定の対応ストリームであるとする。また、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ものとする。

20

【0105】

まず、ストリームの構成や連続領域の構成を決定する(S701)。1VUを1GOP15フレームで構成するとしたとき、式(6)、式(7)に、 $R_s = 20\text{Mbps}$ 、 $T_a = 1\text{秒}$ 、 $R_v = 12\text{Mbps}$ 、 $R_a = 256\text{kbps}$ 、 $T_{vmax} = \text{約}0.5\text{秒}$ の条件を代入することで、 $T_e(i)$ の範囲が3秒以上4秒以下として得られる。 $T_{vmax} = \text{約}0.5\text{秒}$ でこの条件を満たすのは $T_e(i) = 3\text{秒}$ のときとなり、6個のVU毎にCAが挿入されることになる。

【0106】

このときのCAの領域サイズは、3秒分のオーディオデータにパックヘッダやパケットヘッダがつくことを考慮して決定する。以上より、上記S701の処理は、AVデータであるオリジナルストリーム及びその関連データであるアフレコデータを所定の間隔毎に部分AVデータ(CU:すなわち、6個のVU)および部分関連データ(CA)に分割する第1のステップに相当する。

30

【0107】

6個のVUと1個のCAとを連続的に記録可能な空き領域を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(S702)。

【0108】

次に、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(S703)。また、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(S704)、蓄積されている間、S705からS708の処理を繰り返す。

40

【0109】

すなわち、記録用バッファに1ECCブロック分以上のデータが蓄積されていれば、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(S705)。空きがあれば、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(S706)。空きがなければ、9個のVUとCAを記録可能な連続的な空き領域を探して(S707)、その空き領域の先頭へピックアップを移動し(S708)、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

50

する (S 7 0 6) 。

【 0 1 1 0 】

以上より、上記 S 7 0 4 の処理は、一続きの部分 A V データと部分関連データとを記録するための連続領域である第 1 の連続領域を確保する第 2 のステップに相当する。また、上記 S 7 0 6 の処理は、第 1 の連続領域に前記部分 A V データと前記部分関連データとを連続的に記録する第 3 のステップに相当する。

【 0 1 1 1 】

一方、記録用バッファ 1 1 1 に 1 E C C ブロック分以上の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 (S 7 0 9) 、記録終了でなければ S 7 0 4 に移行する。

10

【 0 1 1 2 】

S 7 0 9 において記録終了が指示されていた場合は、以下の処理を実行する。まず、記録用バッファ中の 3 2 K B 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 3 2 K B にする (S 7 1 0) 。次に、そのデータをディスク上に記録する (S 7 1 1 ~ S 7 1 4) 。尚、上記 7 1 1 ~ S 7 1 4 の処理は、S 7 0 5 ~ S 7 0 8 の処理と同様の処理である。

【 0 1 1 3 】

さらに、R A M 1 0 2 上のオリジナルストリームに関する管理情報およびアフレコデータに関する管理情報をそれぞれオリジナルストリーム管理情報ファイルおよびアフレコデータ管理情報ファイルに記録する (S 7 1 5) 。また、ファイルシステム管理情報を光ディスク 1 0 6 に記録する (S 7 1 6) 。なお、その際のファイルシステム管理情報は、C A と C U とが別のファイルとして扱われるように構成する。

20

【 0 1 1 4 】

以上より、上記 S 7 1 6 の処理は、前記部分 A V データおよび前記部分関連データとをそれぞれ別のファイルとして管理すると共に、前記部分 A V データおよび前記部分関連データと前記第 1 の連続領域とを別のファイルとして扱うための情報を管理するファイルシステム管理情報を前記記録媒体上に記録する第 4 のステップに相当する。

【 0 1 1 5 】

また、上記 S 7 1 5 の処理は、前記部分 A V データの再生開始時刻と、前記第 1 の連続領域の位置情報の対応情報とを前記記録媒体に記録する第 5 のステップに相当する。

30

【 0 1 1 6 】

以上の処理と並行するオーディオエンコーダ 1 1 7 、ビデオエンコーダ 1 1 8 およびマルチプレクサ 1 1 3 の動作について説明する。それぞれのエンコーダがエンコードした結果は、オーディオ記録用バッファ 1 1 9 およびビデオ記録用バッファ 1 2 0 に送られる。マルチプレクサ 1 1 3 はそれらのデータを MPEG-2 PS に多重化して記録用バッファ 1 1 4 に格納する。

【 0 1 1 7 】

記録用バッファ 1 1 4 に 1 V U 分のデータが送られ、なおかつ、その V U が $9 \times i$ 番目 (i は 0 以上の整数) の V U であれば、上述のサイズを持った C A を先に記録用バッファ 1 1 1 に送る。

40

【 0 1 1 8 】

さらに、ホスト C P U 1 0 1 に 1 V U 分のデータがエンコードできたことを通知し、ホスト C P U 1 0 1 は V U の先頭 P T S 、パック数および C A を構成するパック数を基に R A M 1 0 2 上のオリジナルストリームに関する管理情報およびアフレコデータに関する管理情報を更新する。

【 0 1 1 9 】

< 再生時の処理 >

すでにアフレコを行ったプログラムに対してユーザから再生が指示された場合の処理を、図 1 7 のフローチャートに沿って説明する。ここで、すでに再生の対象となるプログラム情報ファイルは R A M 1 0 2 に読み込まれているものとする。

50

【 0 1 2 0 】

まず、プログラム情報ファイル中のscene_info()のsc_filenameを参照し、このプログラムが参照しているオリジナルストリームファイルおよびアフレコデータファイルをopenする。同時にそれらを管理するオリジナルストリーム管理情報ファイルを読み込む(S 9 0 1)。

【 0 1 2 1 】

次にシーン番号に0をセットし(S 9 0 2)、シーン番号がscene_table()中のnumber_of_sceneより小さい間(S 9 0 3)、以下のシーン番号に対応するscene_infoの内容を参照して後述するシーンの再生を行い(S 9 0 4)、終わったらシーン番号をインクリメントする(S 9 0 5)。

10

【 0 1 2 2 】

次にシーン再生処理について図18を用いて説明する。まず、scene_info()中のsc_filenameに対応する、すでにRAM 1 0 2に読み込まれたオリジナル管理情報のvideo_unit_table()を参照し、sc_start_PTS以下でなおかつ最大のVU_PTSを持つ、video_unit_info()を探す(S 8 0 1)。上記S 8 0 1の処理は、再生が開始されるシーンのVU番号を求めることとなる。なお、video_unit_table()中でvideo_unit_info()の順番をVU番号と呼ぶ。

【 0 1 2 3 】

次に、continuous_area_table()を参照し、sc_start_PTS以下でなおかつ最大のCA_PTSを持つcontinuous_area_info()を探す(S 8 0 2)。上記S 8 0 2の処理は、再生が開始されるシーンに対応するCAのアドレスを求めることとなる。さらに、その中のCA_PNで指定されるパックから、次のcontinuous_area_info()のCA_PNで指定されるパックの直前のパックまでをアフレコデータファイルから読み出す(S 8 0 3)。

20

【 0 1 2 4 】

次に現在のVU番号に対応するvideo_unit_info()のVU_PNを参照しVUのアドレスを求め(S 8 0 4)、それを基にオリジナルストリームファイルからVUを読み出す(S 8 0 5)。次に、そのシーンの終わりかどうかを判断する(S 8 0 6)。具体的には、現在のシーンの既再生時間がscene_info()中のsc_duration以上であればそのシーンの終わりとする。

【 0 1 2 5 】

シーンの再生が終了していなければ、次にVU番号をインクリメントし(S 8 0 7)、video_unit_info()中のfirst_unit_flagを参照することで、そのvideo_unit_info()が管理するVUがCUの先頭であるか否かが判断される(S 8 0 8)。

30

【 0 1 2 6 】

このときfirst_unit_flagが1であれば、そのvideo_unit_info()が管理するVUはCUの先頭であると判断し、前述の手順で対応するCAのアドレスを求め(S 8 0 9)、アフレコデータファイルからCAの読み出しを行う(S 8 1 0)。一方、first_unit_flagが0であれば、そのvideo_unit_info()が管理するVUはCUの先頭でないと判断し、S 8 0 4 ~ S 8 0 8の処理を繰り返す。

【 0 1 2 7 】

上記のストリームおよびデータの光ディスク1 0 6からの読み出しと平行して、デコード処理を以下のように行う。まず読み出されたVUは、デマルチプレクサ1 1 2に送られ、ビデオPESパケットおよびオーディオPESパケットが取り出され、ビデオPESパケットはビデオ再生用バッファ1 1 1に送られ、オーディオPESパケットはオーディオ再生用バッファ1 1 0に送られる。

40

【 0 1 2 8 】

デマルチプレクサ1 1 2はパックヘッダからSCRを取り出し、システムクロック1 0 5を更新する。ビデオデコーダ1 1 6およびオーディオデコーダ1 1 5は、システムクロック1 0 5が、PESパケットヘッダに付随するタイムスタンプと一致した時点でデコードや出力を行う。

50

【 0 1 2 9 】

本実施の形態 1 においては、オリジナルストリームを格納する C U と、同期再生を行うアフレコデータを格納する C A がディスク上で物理的な近傍にあるため、シーン開始が C U の終端付近の V U からであったとしても、C A から V U のシークによって生じるデータ読み出しの停止時間はわずかで済む。

【 0 1 3 0 】

それに対し、アフレコデータを同期再生するオリジナルストリームの近傍に配置しない場合、シーンの先頭部分におけるアフレコデータの読み出しと、オリジナルストリームの読み出しとの間に生じるシーク時間は最悪ディスクの最内周から最外周に至るものになる。したがって本実施形態に比べ、シーン間で再生が途切れる可能性が高くなる。

10

【 0 1 3 1 】

< アフレコ時の処理 >

次に、ユーザからアフレコが指示された場合の処理について説明する。アフレコ時の処理は前述の再生時の処理にいくつかの処理が加わったものであるため、差異部分についてのみ説明する。

【 0 1 3 2 】

まず、アフレコデータの記録のために、シーンの再生開始と同時にオーディオエンコーダ 1 1 7 を起動し、アフレコデータをエンコードした結果は P E S パケットの形式でオーディオ記録用バッファ 1 1 9 に送る。マルチプレクサ 1 1 3 は P E S パケットをパック化し、記録用バッファ 1 1 4 に送る。その際、パックヘッダの S C R およびパケットヘッダの P T S はオリジナルストリームに合わせる。

20

【 0 1 3 3 】

現在デコード中の C U の範囲を超える P T S を持つパックが記録用バッファ 1 1 4 に到着した時点で、記録用バッファ 1 1 4 に存在するパック列をアフレコデータファイルに記録する。記録対象の C A の位置は現在デコード中の C U の P T S から `continuous__area__table()` を参照することで求める。

【 0 1 3 4 】

C A 記録時にディフェクトに遭遇した場合、記録中の C A は破棄し、別の領域に新規に C A を記録する。これは、記録中の C A の記録領域はディフェクトによって減少し、対応する C U の再生時間分のデータを C A に記録できなくなってしまうためである。その際に、その C A を管理する `continuous__area__info()` 中の `placement__flag` を 0 に変更することで、対応する C U の直前に C A が存在しないことがわかるようにしておく。もちろん、ファイルシステム管理情報における破棄した C A の `extent` は、新規作成した C A の `extent` に置き換える。

30

【 0 1 3 5 】

これにより、非破壊編集時や非破壊編集結果の再生時に、C A と C U とが連続して記録されていない区間が前記 `placement__flag` を参照するだけでわかり、その区間は再生が途切れる可能性が高いことを事前にユーザに伝えることが可能である。また、このフラグを頼りに連続して記録されていない C A と C U とを後から連続して記録されるよう再配置することも可能である。

40

【 0 1 3 6 】

< 実施の形態 1 における変形例 >

本実施の形態 1 においては、アフレコデータファイルにオリジナルストリームファイルと同様に MPEG-2 PS 形式で記録しているが、Elementary Stream、すなわちパック化やパケット化せずに記録することも考えられる。これにより、C A の一部分のみのアフレコデータを書き換える際に、パックから A A U を抜き出して A A U を置き換えて再度パック化する、という手間が不要になる。

【 0 1 3 7 】

また、本実施の形態 1 においては、C A にオーディオデータを記録しているが、他の種類のデータを記録してもよい。例えば、オリジナルストリーム中のビデオに重畳して表示

50

するグラフィックスデータを記録してもよい。

【0138】

また、本実施の形態1においては、1個のAAUが複数のパックにまたがるように記録できるが、またがらないように制限してもよい。このことによって、CAの一部のみのアフレコデータを書き換える際に、書き換えたいAAUの含まれるパックを上書きするだけで済む。

【0139】

また、本実施の形態1においては、アフレコ時にCA中にディフェクトを検出した場合、そのCAを破棄し別の領域に記録している。しかし、ディフェクトがあることを想定し、初期録画時にディフェクト用にマージンを考慮してCAのサイズを決定し、アフレコ時にディフェクトを検出した場合は、そのCA中の次の位置に記録するようにしてもよい。このことにより、CAとCUとが連続的に記録できる。

10

【0140】

また、本実施の形態1においては、別ファイルであるCAとCUとを対応付けるため、CU中のデータの先頭のタイムスタンプからCUおよびCAの先頭アドレスを算出できるようにしているが、対応付けがわかるようになっていればどのような表現形態であってもよいことは言うまでもない。

【0141】

また、本実施の形態1においては、MPEG-2 PSを用いているが、MPEG-2 TSでも同様に実現できることは言うまでもない。

20

【0142】

〔実施の形態2〕

本発明の第2の実施形態について、図19を用いて説明する。

【0143】

本実施の形態2における実施の形態1との違いは、実施の形態1が、同期再生を行う複数のデータを記録媒体上で連続的に配置し、それぞれのデータを別のファイルとして管理するというものであるのに対し、本実施の形態2は、それぞれのデータは同じ再生時間軸上にあるものの同時には再生せず切り替えて再生することを前提にしている点にある。

【0144】

具体的には、DVD-Videoにおけるマルチアングル機能と呼ばれる、同じ時間軸における複数のアングルからの映像を再生中に切り替え可能な機能を想定したものである。

30

【0145】

尚、本実施の形態2に係る記録動作は、上記実施の形態1に係る記録動作において同期再生されるべきオリジナルストリームとアフレコデータとの関係を、同じ再生時間軸上にある2種類のオリジナルストリームの関係に置き換えただけのものであり、実質的な動作は同じである。

【0146】

<ファイル構成>

ビデオ・オーディオデータをMPEG-2 PS形式で多重化し、アングル毎に別ファイルに記録する。図19(a)の例では、第1のアングルからのデータをANGL0001.M2P、第2のアングルからのデータをANGL0002.M2Pに記録する。

40

【0147】

<ディスクへの配置>

図19(b)に示すように第1のアングルからのデータであるANGL0001.M2Pを部分データ2021、2022、2023に分割し、第2のアングルからのデータであるANGL0002.M2Pを部分データ2011、2012、2013に分割し、ディスク2001上で交互に配置する。なお、分割の単位の決定方法は、DVD-Videoにおけるマルチアングルデータの配置と同様であるため、説明を省略する。

【0148】

このことによって、DVD-Videoにおけるマルチアングル切り替えと同様のアングル切り

50

替えのレスポンスを実現すると同時にそれぞれのデータファイルは一般的なMPEG-2 PS対応デコーダで再生可能とすることができる。

【0149】

〔実施の形態3〕

本発明の第3の実施形態について、図21ないし図28を用いて説明する。

【0150】

本実施の形態3における実施の形態1との違いは、実施の形態1では、アフレコ用領域を1個のファイル(すなわち、図4におけるアフレコデータファイル(SHRP0001.PRE))で管理しているのに対し、本実施の形態3では、空き領域を確保しておくためのファイルと個々のAVデータを記録するためのファイルを別にしている点にある。なお、本実施の形態3は実施の形態1と類似するため、相違点に絞って説明を行う。

10

【0151】

<ファイル・ディレクトリ構成>

本実施の形態3におけるファイル・ディレクトリ構成を図21に示す。このファイル・ディレクトリ構成では、実施の形態1におけるファイル・ディレクトリ構成(図4参照)に対して、アフレコ領域予約ファイル(SHRP0001.RSV)、アフレコデータ管理情報ファイル(SHRP0001.PMI)、およびグラフィックスファイル(SHRP0001.PNG)が追加されている。

【0152】

アフレコ領域予約ファイル(SHRP0001.RSV)は、アフレコ用の領域を予約するためのファイルである。アフレコデータ管理情報ファイル(SHRP0001.PMI)は、アフレコデータファイルに対応する管理情報である。グラフィックスファイル(SHRP0001.PNG)は、ビデオに重畳表示するグラフィックスデータを格納するファイルである。また、プログラム情報ファイル(SHRP0001.PGM)、オリジナルストリーム管理情報ファイル(SHRP0001.OMI)、オリジナルストリームファイル(SHRP0001.M2P)は、実施の形態1における同名称のファイルと同じものである。

20

【0153】

アフレコ領域予約ファイルは、録画時にオリジナルストリームファイル1個につき1個記録される。アフレコデータ管理情報ファイルは、アフレコデータファイル1個につき1個作成される。グラフィックスファイルは、録画後に非破壊編集によって追加されるファイルであり、ビデオに重畳表示するタイトルや手書き文字の画像を格納する。グラフィックスファイルにおける画像の格納フォーマットにはPNG(Portable Network Graphics)を用いる。

30

【0154】

また、アフレコデータファイル(SHRP0001.PRE)については、実施の形態1とは異なり、アフレコを行うことによってはじめて生成される。つまり録画時には、実施の形態1におけるアフレコデータファイルの代わりにアフレコ領域予約ファイルが記録されることになる。

【0155】

<AVストリームの形態>

AVストリームの形態については、図5を参照して説明した実施の形態1の構成と同様である。

40

【0156】

<アフレコ領域予約ファイル>

アフレコ領域予約ファイルは、実施の形態1において図6で示されるアフレコデータファイルと同様の構成をとる。すなわち、アフレコ領域予約ファイルは、整数個のContinuous Area(CA)で構成され、CA1個はオリジナルストリームファイルにおけるCU1個に対応して存在し、対応するCUのアフレコデータを記録するための領域を確保する。ただし、ここでのCAはあくまでも領域を確保するためのものであり、再生対象のAVデータではないため、中身のデータは何であってもよい。

【0157】

50

< ディスクへの配置 >

本実施の形態3における各ファイルのディスクへの配置形態について説明する。図22(a)~22(b)は、録画後、アフレコが行われていない状態(すなわち、アフレコデータファイルが作成されていない状態)のファイル配置をしめしている。図22(a)に示す互いに対応するオリジナルストリームファイル(SHRP0001.M2P)とアフレコ領域予約ファイル(SHRP0001.RSV)とは、光ディスク106でCUの直前に対応するCAが配置されるように記録する(図22(b))。

【0158】

次に、オーディオおよびグラフィックスを追加記録した後の、アフレコ領域予約ファイル、アフレコデータファイルおよびグラフィックスファイルのディスクへの配置について図23(a)~23(b)を用いて説明する。図23(a)に、追加記録されるグラフィックスファイル(SHRP0001.PNG)およびアフレコデータファイル(SHRP0001.PRE)の構成を示す。グラフィックスファイルはグラフィックスデータIMGを格納している。アフレコデータファイルは、図22(a)に示したオリジナルストリームファイルにおけるCU#n-1、CU#n、CU#n+1のそれぞれに対応するアフレコオーディオデータPR#1、PR#2、PR#3を格納している。

【0159】

上述のIMGおよびPRのデータは、図23(b)で示す形態にて光ディスク106に配置される。すなわち、図22(b)において、CA#n-1で確保されていた領域にPR#1、CA#nで確保されていた領域にPR#2、CA#n+1で確保されていた領域にPR#3が配置される。また、グラフィックスデータIMGは、CA#n+1で確保されていた領域に配置されている。

【0160】

このように、CAによって確保していた領域にIMGおよびPRのデータを配置することによって、アフレコ領域予約ファイル(SHRP0001.RSV)を構成するCA#n-1、CA#n、CA#n+1の領域サイズが縮小される。この縮小は、ファイルシステム管理情報中におけるそれぞれのCAの領域を管理しているextentを変更することで実現される。

【0161】

このように、アフレコ領域の空きを管理するファイルであるアフレコ領域予約ファイルを導入することで、複数種類のデータを後から追加していくのが容易になる。また、グラフィックスデータとアフレコオーディオデータを別のファイルに格納することによって、グラフィックスデータだけを別のプログラムから参照したりすることができるようになり、柔軟性が増す。

【0162】

< 管理情報ファイルフォーマット >

オリジナルストリーム管理情報ファイルのフォーマットは実施の形態1と同一であり、ここでは説明を省略する。アフレコデータ管理情報ファイルのフォーマットは、実施の形態1におけるオリジナルストリーム管理情報ファイルとほぼ共通であるが、p_attribute()およびcontinuous__area__table()が存在しない点で異なる。

【0163】

次に、プログラム情報ファイルの構成を図24に示す。本実施の形態3に係るプログラム情報ファイルは、録画後から追加されたオーディオデータおよびグラフィックスデータをそれぞれ管理するsubaudio__table()およびgraphics__table()が追加された点で実施の形態1(図14参照)とは異なる。

【0164】

subaudio__table()は、図25(a)に示すように、オーディオデータの個数を表すnumber__of__subaudioと、各オーディオデータに関する情報を格納するsubaudio__info()とで構成される。subaudio__info()は、図25(b)に示すように、所定のオーディオデータを管理するアフレコデータ管理情報ファイルのファイル名を格納するSA__filenameと、所定のオーディオデータの各種属性を管理するSA__flagsと、プログラム中の再生開始タ

10

20

30

40

50

イミングと再生継続時間とをそれぞれ示すSA__start__timeとSA__durationとで構成される。

【 0 1 6 5 】

一方、graphics__table()は、図 2 6 (a) に示すように、グラフィックスファイルの個数を表すnumber__of__graphicsと各グラフィックスファイルに関する情報を格納するgraphics__info()とで構成される。graphics__info()は、図 2 6 (b) に示すように、所定のグラフィックスファイルのファイル名を格納するgr__filenameと、所定のグラフィックスデータの各種属性を管理するgr__flagsと、プログラム中の再生開始タイミングと再生継続時間とをそれぞれ示すgr__start__timeとgr__durationとで構成される。

【 0 1 6 6 】

SA__flagsとgr__flagsとは同一の構成を持ち、図 2 7 (a) に示すようにフラグinterleave__flagを含む。interleave__flagは、1 b i t の情報であり、図 2 7 (b) に示すように、0 b の場合、管理するオーディオデータあるいはグラフィックスファイルが対応する(同期再生する)CUの直前にないことを意味し、1 b の場合、対応する(同期再生する)CUの直前にあることをそれぞれ意味する。このフラグを参照することで、非破壊編集結果を再生する際に再生が途切れる可能性を知ることが可能である。すなわち、このフラグが0 bであれば、CAへのシークが発生し、再生が途切れる可能性が高いことがわかる。

【 0 1 6 7 】

< CU単位決定方法 >

CU単位決定方法については、図 7 および図 8 を参照して説明した実施の形態 1 の方法と同様である。

【 0 1 6 8 】

< 記録時の処理 >

記録時の処理については、図 1 6 を参照して説明した実施の形態 1 の処理と同様である。

【 0 1 6 9 】

< 再生時の処理 >

すでにアフレコを行ったプログラムに対してユーザから再生が指示された場合の処理を以下に説明する。基本的な流れは実施の形態 1 においてフローチャート図 1 7 に沿って説明したとおりであるため同一の処理を行う部分については説明を省略する。ただし、本実施の形態 3 に係る再生処理では、シーン再生処理が実施の形態 1 とは異なるため、シーン再生処理のみに絞って図 2 8 に沿って説明する。

【 0 1 7 0 】

まず、scene__info()中のsc__filenameに対応する、すでにRAM 1 0 2 に読み込まれたオリジナル管理情報のvideo__unit__table()を参照し、sc__start__PTS以下でなおかつ最大のVU__PTSを持つ、video__unit__info()を探す(S 8 0 1)。なお、video__unit__table()中でvideo__unit__info()の順番をVU番号と呼ぶ。

【 0 1 7 1 】

次に、プログラム情報ファイルのgraphics__table()およびsubaudio__table()を参照し、現在のVUを含むCUとの同期再生を行うグラフィックスファイルおよびオーディオデータが存在するかどうかを検索する(S 8 0 2')。そのようなグラフィックスファイルおよびオーディオデータが存在した場合、グラフィックスファイルについてはそのファイルの読み出しを行い、オーディオデータについては、そのオーディオデータを管理するオーディオデータ管理情報ファイルを参照し、対応するオーディオデータの読み出しを行う(S 8 0 3')。

【 0 1 7 2 】

次に現在のVU番号に対応するvideo__unit__info()のVU__PNを参照しVUのアドレスを求め(S 8 0 4)、それを基にオリジナルストリームファイルからVUを読み出す(S 8 0 5)。次に、そのシーンの終わりかどうかを判断する(S 8 0 6)。具体的には、現

10

20

30

40

50

在のシーンの既再生時間がscene__info()中のsc__duration以上であればそのシーンの終わりとする。

【 0 1 7 3 】

シーンの再生が終了していなければ、次にV U番号をインクリメントし(S 8 0 7)、video__unit__info()中のfirst__unit__flagを参照する。このときfirst__unit__flagが1であれば、そのvideo__unit__info()が管理するV UはC Uの先頭であると判断し(S 8 0 8)、同期再生を行うグラフィックスファイルおよびオーディオデータの存在を前述の手順で確認し(S 8 0 9)、存在した場合には前述の手順で読み出す(S 8 1 0)。

【 0 1 7 4 】

< アフレコ時の処理 >

ユーザからアフレコが指示された場合の処理について説明する。アフレコ時の処理は前述の再生時の処理にいくつかの処理が加わったものであるため、差異部分についてのみ説明する。

【 0 1 7 5 】

まず、記録媒体においてアフレコするための領域が存在するかどうかを確認される。具体的には、アフレコ対象のストリームに関する管理情報ファイル中のcontinuous__area__info()を調べ、アフレコ領域予約ファイルによって確保された各領域のサイズがアフレコデータを記録可能なサイズかどうかを調べる。もし十分なサイズがあればその領域にアフレコデータを記録し、十分なサイズが無ければ、アフレコデータは、アフレコ領域予約ファイルによって確保された領域ではなく、別の領域に記録することになる。

【 0 1 7 6 】

次に、再生開始と同時にオーディオエンコーダ1 1 7を起動し、エンコードした結果はP E Sパケットの形式でオーディオ記録用バッファ1 1 9に送る。マルチプレクサ1 1 3はPESパケットをパック化し、記録用バッファ1 1 4に送る。その際、パックヘッダのS C RおよびパケットヘッダのP T Sはオリジナルストリームに合わせる。

【 0 1 7 7 】

現在デコード中のC Uの範囲を超えるP T Sを持つパックが記録用バッファ1 1 4に到着した時点で、記録用バッファ1 1 4に存在するパック列をアフレコデータファイルに記録する。記録対象のC Aの位置は現在デコード中のC UのP T Sからcontinuous__area__table()を参照することで求める。C Aによって確保されている領域にアフレコデータをアフレコデータファイルに記録する。

【 0 1 7 8 】

アフレコが終わった時点では以下のことを行う。まず、記録したアフレコデータファイルに対応するアフレコデータ管理ファイルを作成する。その際に、1個のC U毎に1個のvideo__unit__info()を作成する。

【 0 1 7 9 】

次に、プログラム情報ファイル中のsubaudio__tableにエントリを追加する。このとき、アフレコデータをアフレコ領域予約ファイルで確保されていた領域に記録したならば、そのことを示すためSA__flags()のinterleave__flagを1にセットする。そうでなければ0をセットする。

【 0 1 8 0 】

また、アフレコ領域予約ファイルに関して、アフレコデータが記録された領域をファイルの管理対象から外す。すなわちサイズを小さくする。さらに、その分だけcontinuous__area__table()中の各エントリのCA__PNを減ずる。このことによって、continuous__area__info()を参照したとき、各C Aに後どれだけデータを記録できるかがわかる。

【 0 1 8 1 】

< グラフィックスデータ付加時の処理 >

ユーザからビデオに重畳表示するグラフィックスデータの付加が指示された場合の処理について説明する。まず、グラフィックスデータを格納したファイルを記録するための領

10

20

30

40

50

域が存在するかどうか調べる。具体的にはグラフィックスデータ付加対象のストリームに関する管理情報ファイル中のcontinuous_area_info()を調べ、グラフィックスデータを重畳表示を開始するビデオフレームが含まれるCUに対応するCAにグラフィックスデータを記録可能な領域が存在するかどうかを調べる。

【0182】

上記CAにおいてグラフィックスデータの記録が可能であればその領域に記録し、アフレコ終了時と同様、グラフィックスデータを記録した領域をアフレコ領域予約ファイルから開放する。開放した分だけcontinuous_area_info()中の後続するエントリのCA_PNを減ずる。また、プログラム情報ファイルのgraphics_table()にgraphics_info()のエントリを1個追加し、そのエントリ中のgr_flags()のinterleave_flagを1にセットする。この場合、同時再生するビデオデータとディスク上で近接した位置にグラフィックスデータを記録されていることにより、ビデオ再生時にグラフィックスデータを読み出すためのシークが不要となる。そのため、シークによるビデオ再生の途切れや、電力消費等を抑えることが可能になる。

10

【0183】

一方、上記CAにおいてグラフィックスデータの記録が可能でなければ、該グラフィックスデータは別の領域に記録する。また、プログラム情報ファイルのgraphics_table()にgraphics_info()のエントリを1個追加し、そのエントリ中のgr_flags()のinterleave_flagを0にセットする。再生時にはこのフラグを参照することで、ユーザにビデオ再生に途切れが発生するかもしれないことをビデオ再生前に知らせることができる。

20

【0184】

<実施の形態3における変形例>

本実施の形態3では、グラフィックスファイルやアフレコデータファイルは録画後に追加記録しているが、録画時に記録してもよいことは言うまでもない。その場合でも、グラフィックスファイルやアフレコデータファイルをビデオファイルとは独立したファイルとして扱うことができ、ビデオファイルは一般的なMPEG-2 PSファイルであり、なおかつ同期再生にあたってシークが必要ない。

【0185】

また本実施の形態3では、グラフィックスファイルにPNGファイルフォーマットを用いているが、JPEG等他のファイルフォーマットを用いてもよいことは言うまでもない。

30

【0186】

〔実施の形態4〕

本発明の第4の実施形態について、図29ないし図30を用いて説明する。本実施の形態4の位置付けは、実施の形態1におけるディスクへの配置およびCU単位決定方法のバリエーションである。したがって、それ以外の部分については共通するため、相違点に絞って説明を行う。

【0187】

<ファイル・ディレクトリ構成>

実施の形態1と同一であるため説明を省略する。

【0188】

<AVストリームの形態>

実施の形態1と以下の点を除き同一である。実施の形態1と異なる点は、本実施形態においてはCUが連続的に記録されなくてもよい点である。

40

【0189】

<アフレコデータファイル>

実施の形態1と同一であるため説明を省略する。

【0190】

<ディスクへの配置>

オリジナルストリームファイルおよびアフレコデータファイルのディスクへの配置について図29を用いて説明する。図29(a)に示す、互いに対応するオリジナルストリー

50

ムファイル(SHRP0001.M2P)とアフレコデータファイル(SHRP0001.PRE)とは、原則的には実施の形態1のように光ディスク106でCUの直前に対応するCAが配置されるように記録するが、実施の形態1と異なりCUの途中で分断することを許す。図29(b)のCU#n-1とCU#nがその例である。ただしCAに関しては実施の形態1と同様、1個のCA内での分断は許さない。また、1個の連続領域中に含まれるVUの再生時間の合計はCUの再生時間以上でなければならない。

【0191】

このように、CUの途中における分断を許すことによって、空き領域を有効活用できる可能性が高まる。例えば、CUが16秒でなかつ20秒分の連続した空き領域が光ディスク106上にある場合、CUの途中で分断を許さなければ前記空き領域に16秒しか記録できず、残りの4秒は無駄になるが、CUの途中で分断できれば20秒分記録することが可能である。

10

【0192】

<CU単位決定方法>

CU再生時間の決定方法について、図30を用いて説明する。この決定方法では、実施の形態1と同様、機器間での互換性確保のため、基準となるデバイス(リファレンス・デバイス・モデル)と基準となるアフレコアルゴリズム(リファレンス・アフレコ・アルゴリズム)とを想定し、次にそれらを用いてアフレコを行った際にシームレス再生が破綻しないようにCU再生時間を決める。

【0193】

リファレンス・デバイス・モデルについては実施の形態1と同様であるため説明を省略する。

20

【0194】

リファレンス・アフレコ・アルゴリズムは以下の(a)、(b)で表される。(a)原則的には、現在のCUの読み出しが終了したときにCAの記録を実行する。(b)ただし、現在読み出し中のCUの、次のCUの末尾が別の連続領域に含まれていた場合、CAの記録は現在読み出し中のCUの、次のCUの読み出しが終了するときまで延期する。

【0195】

図30(a)、30(b)に例を示す。図30(a)は上記の(a)の一例である。尚、図30中の(1)から(8)までの番号は、以下の説明中の(1)から(8)までの番号に対応する。

30

【0196】

(1)N番目のCUであるCU#Nを読み出す。(2)次に記録すべきCAであるCA#Mのアフレコデータのエンコードが終了するのを待ち、終了したらCA#Mへピックアップを移動。(3)CA#Mにアフレコデータを記録。(4)CU#N+1へピックアップを移動。(5)CU#N+1を読み出し。(6)次に記録すべきCAであるCA#M+1のアフレコデータのエンコードが終了するのを待ち、終了したらCA#M+1へピックアップを移動。(7)CA#Mにアフレコデータを記録。(8)CU#N+2へピックアップを移動。

【0197】

図30(b)は上記の(b)の一例である。(1)まずCU#Nを読み出す。このとき、次のCUであるCU#N+1の終端が別の連続領域に存在するため、CU#Nの読み出し直後にCAの記録は行わずCU#N+1のうち同じ連続領域に存在するCU#N+1の前半部まで読み出す。(2)CU#N+1を連続領域の終端まで読み出し終わったら、CA#Mへのピックアップの移動を行い、(3)CA#Mにアフレコデータを記録し、(4)CU#N+1の残りの部分へのピックアップの移動を行う。

40

【0198】

このようなアルゴリズムを用いることによって、CUの途中で分断していた場合でも連続領域間のジャンプとCA記録のためのジャンプを共通化でき、ジャンプによるデータ読み出しの途切れを最小化し、その結果、連続領域長およびCU・CAの単位を小さくすることが出来る。このことによってディスク上での空き領域の有効利用が図れる。

50

【 0 1 9 9 】

前記リファレンス・デバイス・モデルにおいて、前記リファレンス・アフレコ・アルゴリズムを用いてアフレコを行った場合、次のような条件を満たせば、アフレコ用バッファ 504 のオーバーフローおよびトラックバッファ 502 のアンダーフローがないことが保証できる。

【 0 2 0 0 】

その条件とは、実施の形態 1 と同様式 (1) が成立することである。なお本実施の形態 4 における記号は、特に説明がない限り実施の形態 1 と同じ意味を表す。

【 0 2 0 1 】

式 (1) 中の $T_w(i)$ は実施の形態 1 の式 (4) と同様であるが、 $T_r(i)$ については式 (8) のように実施の形態 1 の式 (3) から T_a を除いた形態、すなわちジャンプを含めない形態となる。

10

【 0 2 0 2 】

【数 8】

$$Tr(i) = Te(i) \times Ro/Rs + Te(i) \times Ra/Rs \quad \dots(8)$$

【 0 2 0 3 】

ジャンプを含めない理由は以下のとおりである。本実施の形態 4 においては C U 読み出し途中のジャンプ時に C A 記録時を行うため、C U 読み出し途中のジャンプは C A 記録時のジャンプとして扱うからである。このことにより最終的な C U の単位および連続領域の単位を小さくすることができる。

20

【 0 2 0 4 】

式 (1) に式 (8) および式 (4) を代入して $Te(i)$ で解くと、リアルタイムアフレコを保証可能な $Te(i)$ の条件、すなわち以下の式 (9) が得られる。

【 0 2 0 5 】

【数 9】

$$Te(i) \geq \frac{2Ta \times Rs}{Rs - Ro - 2Ra} \quad \dots(9)$$

【 0 2 0 6 】

つまり、アフレコ保証可能な C U 再生時間下限値 Te_{min} は、以下の式 (10) に示すものとなる。

30

【 0 2 0 7 】

【数 10】

$$Te_{min} = \frac{2Ta \times Rs}{Rs - Ro - 2Ra} \quad \dots(10)$$

【 0 2 0 8 】

このとき、C U 再生時間の上限値 Te_{max} は、以下の式 (11) のように設定する。ここで、 Tv_{max} は V U の最大再生時間である。

40

【 0 2 0 9 】

【数 11】

$$Te_{max} = \frac{2Ta \times Rs}{Rs - Ro - 2Ra} + Tv_{max} \quad \dots(11)$$

【 0 2 1 0 】

C U 再生時間の上限値を設定するのは、アフレコ用音声と通常音声との同期再生に必要な遅延用メモリの最大量を見積り可能にし、再生互換性を保証するためである。尚、本実施の形態 4 では、オーディオビットレート R_a およびビデオビットレート R_v に応じて多重化間隔下限値 Te_{min} を設定しているが、ビットレートに関わらず一定の下限値を設

50

定しても良い。ただし、その値は最大のビットレートに基づいたものでなければならない。

【0211】

また、CU再生時間が上記の制限を満たせば、ストリーム中のVU再生時間は固定でも可変でも構わない。

【0212】

<必要なバッファメモリ量>

本実施形態においてアフレコ時に必要なトラックバッファ502の容量は以下の考えに基づいて算出すればよい。まず、本実施の形態4において最も容量の必要な場合を想定すると、CAへのアフレコデータ記録が続けて発生する場合である。具体的には、CUの末尾の直前で分断している場合である。言い換えると、CUが2個の連続領域に分割して記録されていて、かつ、そのほとんどのデータが先行する連続領域に存在する場合である。

10

【0213】

この場合、前述のリファレンス・アフレコ・アルゴリズムに従うと、CUの末尾の直前でCAへのアフレコデータ記録を行い、次にこのCUの末尾のわずかなデータを読んだら直ちに次のCAへのアフレコデータ記録を行い、もどってCAの読み込みを行うことになる。これに対応するためにはCAへのアフレコデータ記録2回分および1個のCA読み込みの期間にわたって再生を継続可能な容量が必要となる。具体的には式(12)にしたがってトラックバッファ502の容量Bpbを確保すればよい。

【0214】

【数12】

20

$$Bpb = (2 \times (2 \times Ta + T_{\max} \times Ra / Rs) + T_{\max} \times Ra / Rs) \times Ro \quad \dots (12)$$

【0215】

<管理情報ファイルフォーマット>

実施の形態1と同一であるため説明を省略する。

【0216】

<記録時の処理>

実施の形態1とは、CUをディスク上で連続的に記録する制限以外同一であるため説明を省略する。

30

【0217】

<再生時の処理>

実施の形態1と同一であるため説明を省略する。

【0218】

<アフレコ時の処理>

実施の形態1とは、アフレコの際に図30記載のアルゴリズムを用いる点以外同一であるため説明を省略する。

【0219】

<実施の形態4における変形例>

本実施の形態4は、実施の形態1のバリエーションとして説明したが、第3の実施形態のようにアフレコ領域予約ファイルを用いた場合や従来技術として参照した特開2001-43616号のように同一ファイルを用いた場合にも適用できることは言うまでもない。すなわち、本実施形態で開示した発明の本質は、アフレコ用領域と初期記録したビデオデータの物理的な配置とその配置に関するパラメータを設定するためモデル設定にある。

40

【図面の簡単な説明】

【0220】

【図1】本発明の一実施形態を示すものであり、図1(a)は本発明の実施の形態1におけるオリジナルストリームファイルおよびアフレコデータファイルのデータ構成、図1(b)は上記オリジナルストリームファイルおよびアフレコデータファイルのディスク上の配置を示す図である。

50

【図 2】本発明の実施の形態に係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 3】図 3 (a) はディレクトリ / ファイル構成を示す図であり、図 3 (b) はそのディレクトリ / ファイル構成の U D F における管理情報の関係を示す図である。

【図 4】本発明の実施の形態 1 におけるファイル / ディレクトリ構成を示す図である。

【図 5】図 5 (a) ~ 図 5 (c) は、本発明の実施の形態 1 におけるオリジナルストリームファイルの構成を示す図である。

【図 6】本発明の実施の形態 1 におけるアフレコデータファイルの構成を示す説明図である。

【図 7】本発明の実施の形態 1 におけるリファレンス・デバイス・モデルを示す図である

10

【図 8】本発明の実施の形態 1 におけるリファレンス・アフレコ・アルゴリズムを示す図である。

【図 9】本発明の実施の形態 1 におけるストリーム管理情報ファイルの構成を示す図である。

【図 1 0】図 1 0 (a) ~ 図 1 0 (b) は、本発明の実施の形態 1 における video__unit__table の構成を示す図である。

【図 1 1】図 1 1 (a) ~ 図 1 1 (b) は、本発明の実施の形態 1 における VU__flags の構成を示す図である。

【図 1 2】図 1 2 (a) ~ 図 1 2 (b) は、本発明の実施の形態 1 における continuous__area__table の構成を示す図である。

20

【図 1 3】図 1 3 (a) ~ 図 1 3 (b) は、本発明の実施の形態 1 における C A __flags の構成を示す図である。

【図 1 4】本発明の実施の形態 1 におけるプログラム情報ファイルの構成を示す図である

【図 1 5】図 1 5 (a) ~ 図 1 5 (b) は、本発明の実施の形態 1 における scene__table の構成を示す図である。

【図 1 6】本発明の実施の形態 1 における記録処理の流れを示すフローチャートである。

【図 1 7】本発明の実施の形態 1 における再生処理の流れを示すフローチャートである。

【図 1 8】本発明の実施の形態 1 におけるシーン再生処理の流れを示すフローチャートである。

30

【図 1 9】本発明の他の実施形態を示すものであり、図 1 9 (a) は実施の形態 2 における 2 種類のストリームファイルのデータ構成、図 1 9 (b) はこれらのストリームファイルのディスク上での配置を示す図である。

【図 2 0】従来技術を示すものであり、図 2 0 (a) はストリームファイルのデータ構成、図 2 0 (b) はこのストリームファイルのディスク上での配置を示す図である。

【図 2 1】本発明の実施の形態 3 におけるファイル / ディレクトリ構成を示す図である。

【図 2 2】本発明の一実施形態を示すものであり、図 2 2 (a) は本発明の実施の形態 3 におけるオリジナルストリームファイルおよびアフレコ領域予約ファイルのデータ構成、図 2 2 (b) は録画直後のオリジナルストリームファイルおよびアフレコ領域予約ファイルのディスク上での配置を示す図である。

40

【図 2 3】図 2 3 (a) は本発明の実施の形態 3 におけるグラフィックスファイルおよびアフレコデータファイルのデータ構成、図 2 3 (b) はアフレコ・非破壊編集後のグラフィックスファイル、アフレコデータファイル、オリジナルストリームファイルおよびアフレコ領域予約ファイルのディスク上での配置を示す図である。

【図 2 4】本発明の実施の形態 3 におけるプログラム情報ファイルの構成を示す図である

【図 2 5】図 2 5 (a) ~ 図 2 5 (b) は、本発明の実施の形態 3 における subaudio__table の構成を示す図である。

【図 2 6】図 2 6 (a) ~ 図 2 6 (b) は、本発明の実施の形態 3 における graphics__ta

50

bleの構成を示す図である。

【図27】図27(a)～図27(b)は、本発明の実施の形態3におけるSA_flagsおよびgr_flagsの構成を示す図である。

【図28】本発明の実施の形態3におけるシーン再生処理の流れを示すフローチャートである。

【図29】本発明の一実施形態を示すものであり、図29(a)は本発明の実施の形態4におけるオリジナルストリームファイルおよびアフレコデータファイルのデータ構成、図29(b)は上記オリジナルストリームファイルおよびアフレコデータファイルのディスク上での配置を示す図である。

【図30】図30(a), 30(b)は、本発明の実施の形態4におけるリファレンス・アフレコ・アルゴリズムを示す図である。

10

【符号の説明】

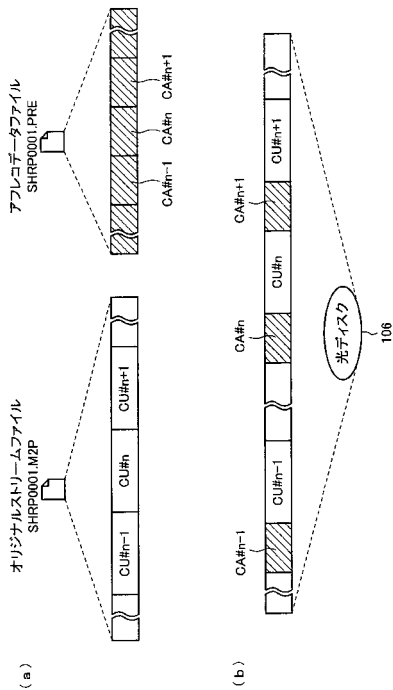
【0221】

- 100 バス
- 101 ホストCPU
- 102 RAM
- 103 ROM
- 104 ユーザインタフェース
- 105 システムクロック
- 106 光ディスク(記録媒体)
- 107 ピックアップ
- 108 ECCデコーダ
- 109 ECCエンコーダ
- 110 オーディオ再生用バッファ
- 111 ビデオ再生用バッファ
- 112 デマルチプレクサ
- 113 マルチプレクサ
- 114 記録用バッファ
- 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
- 119 オーディオ記録用バッファ
- 120 ビデオ記録用バッファ
- 121 デマルチプレクサ
- 122 アフレコデータ再生用バッファ

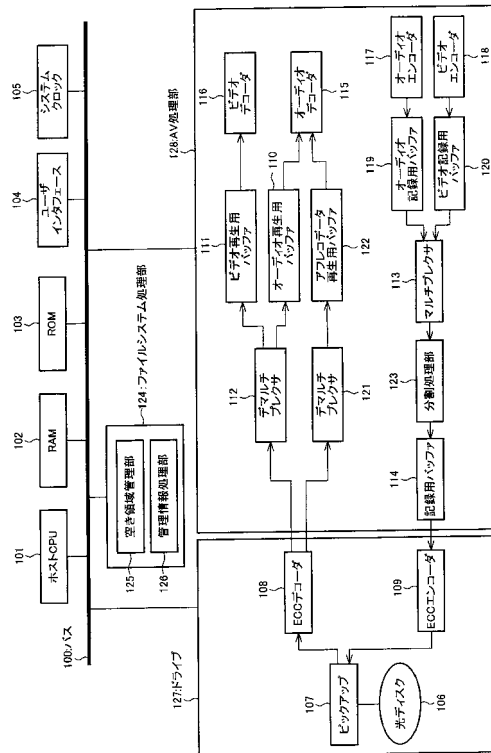
20

30

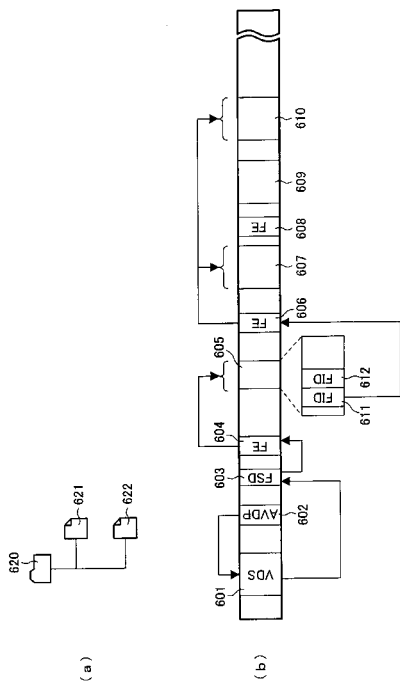
【図1】



【図2】



【図3】



【図4】



【 図 9 】

```

ストリーム管理情報ファイル {
  o_attribute()
  video_unit_table()
  p_attribute()
  continuous_area_table()
}

```

【 図 10 】

```

(a)
video_unit_table() {
  number_of_video_unit
  for (i=0; i<number_of_video_unit; i++){
    video_unit_info()
  }
}

```

```

(b)
video_unit_info() {
  VU_flags
  VU_PTS
  VU_PN
}

```

【 図 11 】

```

(a)  VU_flags(){
      first_unit_flag
}

```

(b)

first_unit_flag	0b	CUの先頭ではない
	1b	CUの先頭である

【 図 12 】

```

(a)
continuous_area_table() {
  number_of_continuous_area
  for (i=0; i<number_of_continuous_area; i++){
    continuous_area_info()
  }
}

```

```

(b)
continuous_area_info() {
  CA_flags
  CA_PTS
  CA_PN
}

```

【 図 13 】

```

(a)  CA_flags(){
      placement_flag
}

```

(b)

placement_flag	0b	対応するCUの直前に配置されていない
	1b	対応するCUの直前に配置されている

【 図 14 】

```

プログラム情報ファイル {
  pg_attribute()
  scene_table()
}

```

【 図 15 】

```

(a)  scene_table() {
      number_of_scene
      for (i=0; i<number_of_scene; i++){
        scene_info()
      }
}

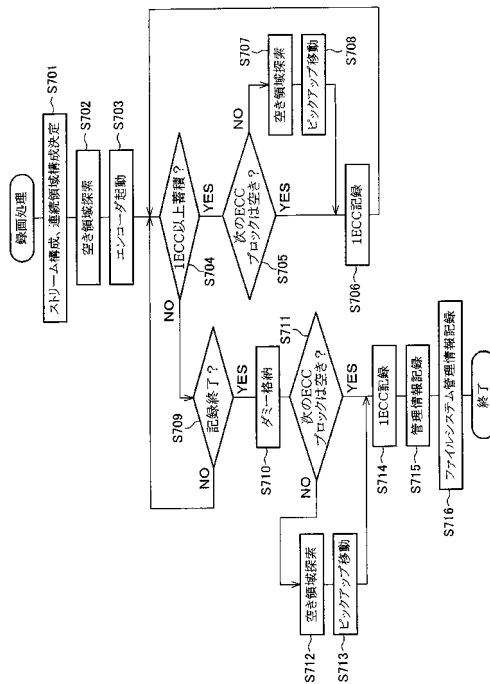
```

```

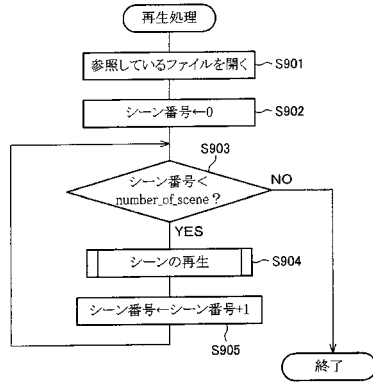
(b)  scene_info() {
      sc_filename
      sc_start_PTS
      sc_duration
}

```

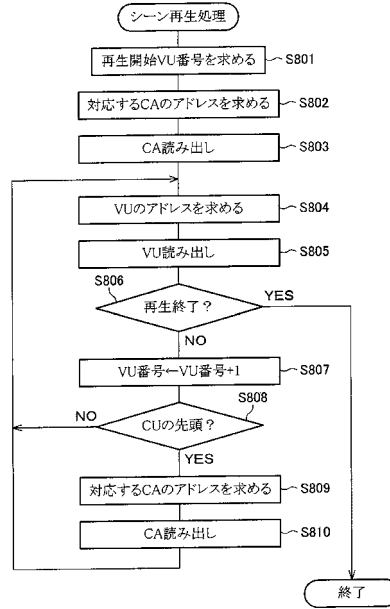
【 図 16 】



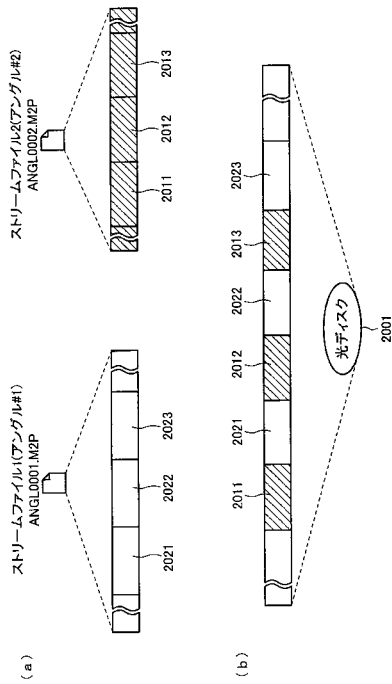
【図17】



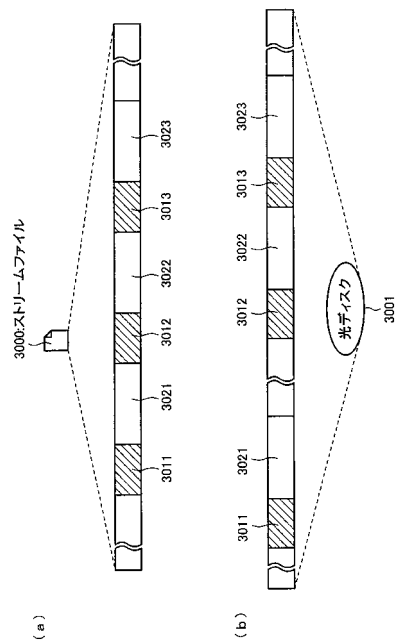
【図18】



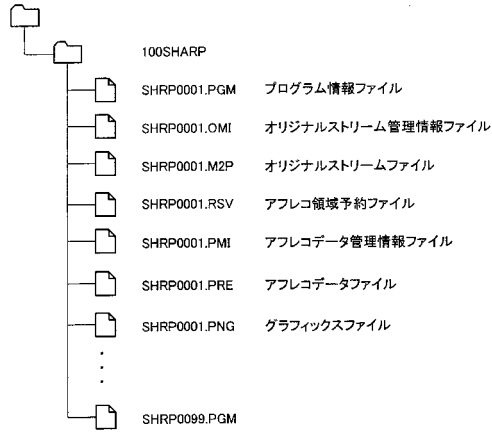
【図19】



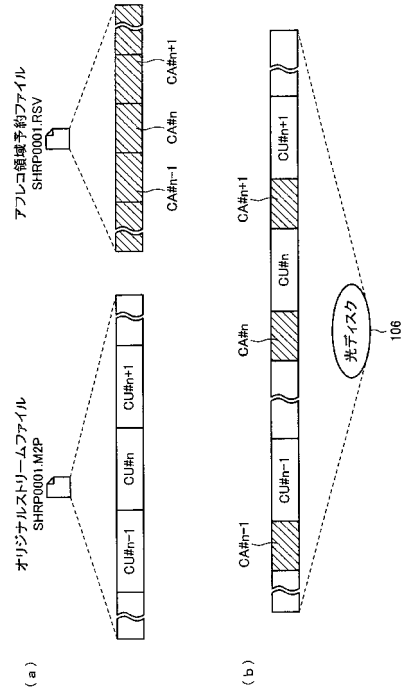
【図20】



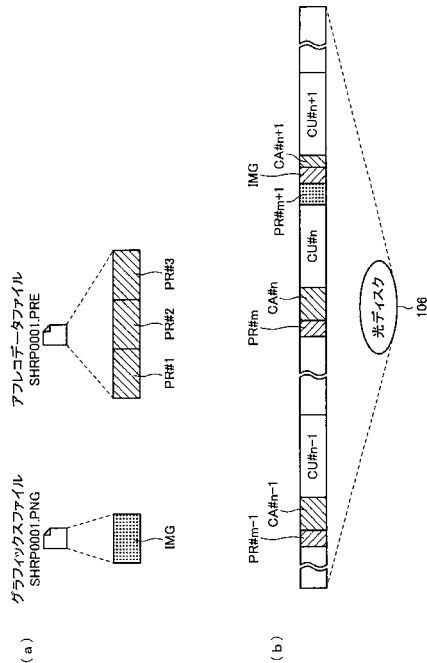
【図 2 1】



【図 2 2】



【図 2 3】



【図 2 5】

```
(a)
subaudio_table() {
    number_of_subaudio
    for (i=0; i<number_of_subaudio; i++){
        subaudio_info()
    }
}

(b)
subaudio_info() {
    SA_filename
    SA_flags
    SA_start_time
    SA_duration
}
```

【図 2 6】

```
(a)
graphics_table() {
    number_of_graphics
    for (i=0; i<number_of_graphics; i++){
        graphics_info()
    }
}

(b)
graphics_info() {
    gr_filename
    gr_flags
    gr_start_time
    gr_duration
}
```

【図 2 4】

```
プログラム情報ファイル {
    pg_attribute()
    scene_table()
    subaudio_table()
    graphics_table()
}
```

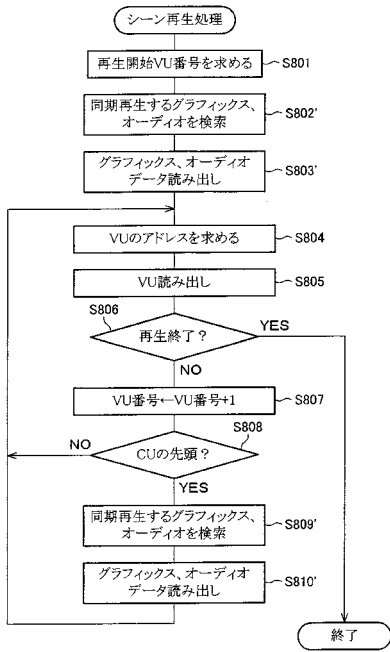
【図 27】

```
(a) SA_flags/gr_flags{
    interleaved_flag
}
```

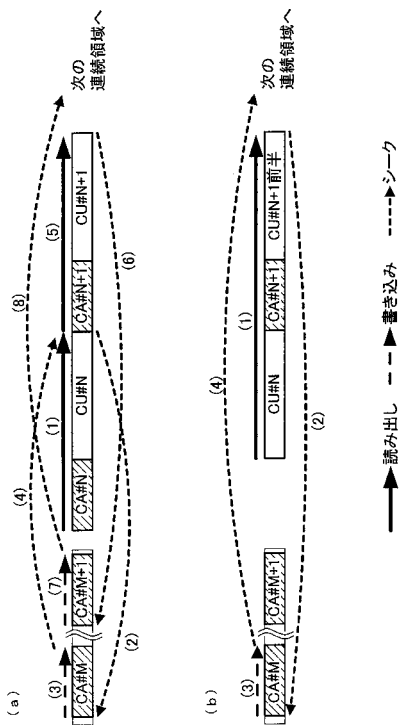
(b)

interleaved_flag	0b	CUの直前に存在しない
	1b	CUの直前に存在する

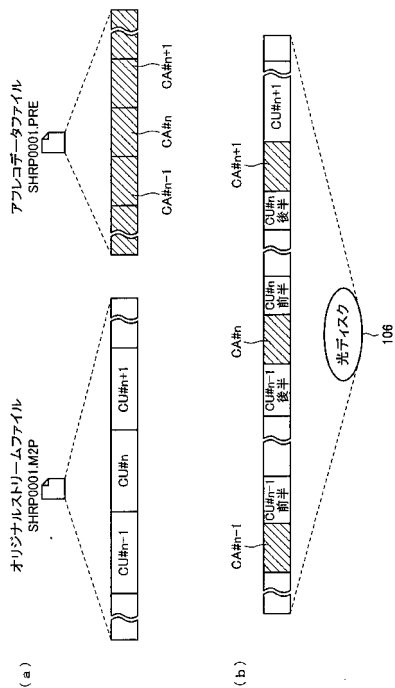
【図 28】



【図 30】



【図 29】



フロントページの続き

(51)Int.Cl. F I
H 0 4 N 5/85 (2006.01) H 0 4 N 5/85 Z

審査官 宮下 誠

(56)参考文献 特開平 1 1 - 2 0 3 7 8 6 (J P , A)
特開平 1 1 - 2 9 8 8 4 5 (J P , A)
特開平 1 1 - 1 4 4 3 7 8 (J P , A)
特開 2 0 0 2 - 2 0 9 1 8 6 (J P , A)
特開 2 0 0 0 - 1 9 5 2 3 1 (J P , A)
特開 2 0 0 2 - 2 4 5 7 5 3 (J P , A)
特開 2 0 0 0 - 3 3 9 8 6 0 (J P , A)

(58)調査した分野(Int.Cl. , D B名)
G 1 1 B 2 7 / 0 0
G 1 1 B 2 0 / 1 0
H 0 4 N 5 / 7 6
H 0 4 N 5 / 9 1