

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4552889号
(P4552889)

(45) 発行日 平成22年9月29日(2010.9.29)

(24) 登録日 平成22年7月23日(2010.7.23)

(51) Int.Cl.

F I

G 1 1 B 27/00 (2006.01)

G 1 1 B 27/00 D

G 1 1 B 20/10 (2006.01)

G 1 1 B 20/10 3 1 1

H O 4 N 5/91 (2006.01)

H O 4 N 5/91 N

H O 4 N 5/85 (2006.01)

H O 4 N 5/91 Z

G 1 1 B 20/12 (2006.01)

H O 4 N 5/85 Z

請求項の数 12 (全 47 頁) 最終頁に続く

(21) 出願番号 特願2006-131909 (P2006-131909)
 (22) 出願日 平成18年5月10日(2006.5.10)
 (65) 公開番号 特開2007-305213 (P2007-305213A)
 (43) 公開日 平成19年11月22日(2007.11.22)
 審査請求日 平成21年4月2日(2009.4.2)

(73) 特許権者 000002185
 ソニー株式会社
 東京都港区港南1丁目7番1号
 (74) 代理人 100082762
 弁理士 杉浦 正知
 (72) 発明者 磯部 幸雄
 東京都品川区北品川6丁目7番35号 ソ
 ニー株式会社内
 (72) 発明者 有留 憲一郎
 東京都品川区北品川6丁目7番35号 ソ
 ニー株式会社内
 (72) 発明者 森本 直樹
 東京都品川区北品川6丁目7番35号 ソ
 ニー株式会社内

最終頁に続く

(54) 【発明の名称】 記録装置、記録方法および記録プログラム、ならびに、撮像装置および撮像方法

(57) 【特許請求の範囲】

【請求項1】

ビデオデータとオーディオデータとを多重化して記録媒体に記録する記録装置において、

ビデオデータおよびオーディオデータが入力されるデータ入力部と、
 上記ビデオデータおよびオーディオデータの記録開始および記録停止の指示が入力される記録指示入力部と、

上記ビデオデータおよびオーディオデータを多重化し、多重化されたストリームをストリームファイルとして記録媒体に記録する記録部と、

上記記録媒体に記録される上記ストリームファイルに対し、少なくとも、該ストリームファイルの再生時刻情報とアドレス情報とを対応付けた属性ファイルと、

該ストリームファイルに対して再生開始点と再生終了点とを設定することにより再生区間を指定する1以上の再生区間データが格納され、上記ストリームファイルに対する再生時刻情報を示すマーク情報が格納可能な再生リストファイルとを生成する管理情報生成部と、

上記記録部および上記管理情報生成部とを制御する制御部とを有し、

上記制御部は、

上記記録指示入力部の指示に基づく記録開始から記録停止までの区間に対応する上記ビデオデータおよびオーディオデータを1の上記ストリームファイルとして上記記録媒体に

10

20

記録するように上記記録部を制御すると共に、該1のストリームファイルの全体を再生区間として指定する上記再生区間データを生成し、該再生区間データが指定する上記ストリームファイルの先頭の位置に対応する上記マーク情報を上記再生リストファイルに格納するように上記管理情報生成部を制御する記録装置。

【請求項2】

請求項1に記載の記録装置において、
上記制御部は、

新たな上記ストリームファイルが上記記録媒体に記録された際には、該ストリームファイルの再生時刻情報とアドレス情報とを対応付けた属性ファイルを生成するように上記管理情報生成部を制御する記録装置。

10

【請求項3】

請求項1に記載の記録装置において、
上記記録指示入力部は、ユーザの操作を受け付ける記録装置。

【請求項4】

請求項3に記載の記録装置において、
上記制御部は、

上記ストリームファイルを上記記録媒体に記録した直後に対応する上記属性ファイルを記録する際には、該属性ファイルの記録が完了するまで上記記録指示入力部に対する上記記録開始の上記操作を受け付けないように制御する記録装置。

20

【請求項5】

請求項1に記載の記録装置において、
上記制御部は、

上記記録指示入力部による記録停止の指示に基づき、記録された上記ストリームファイルを再生区間として指定する上記再生区間データ内に、該ストリームファイルと該ストリームファイルに対して後続するストリームファイルとがフレームタイミングで連続的に再生されることを示す情報を設定して上記再生区間データを上記記録媒体に記録するように制御する記録装置。

【請求項6】

請求項1に記載の記録装置において、
上記制御部は、

上記ストリームファイル内で、上記ビデオデータの先端の再生時刻と上記オーディオデータの先端の再生時刻とが一致し、上記ビデオデータの終端の再生時刻よりも上記オーディオデータの終端の再生時刻が後になるように上記記録部を制御する記録装置。

30

【請求項7】

請求項1に記載の記録装置において、
上記制御部は、

新たな上記ストリームファイルが上記記録媒体に記録された際には、該新たなストリームファイルに対応する上記再生区間データを上記再生リストファイルに追加するように上記管理情報生成部を制御する記録装置。

【請求項8】

請求項7に記載の記録装置において、
上記制御部は、

1の上記記録媒体に対して唯一つの上記再生リストファイルを生成するように制御する記録装置。

40

【請求項9】

ビデオデータとオーディオデータとを多重化して記録媒体に記録する記録方法において、

データ入力部に入力されたビデオデータおよびオーディオデータの記録開始および記録停止の指示が入力される記録指示入力ステップと、

上記ビデオデータおよびオーディオデータを多重化し、多重化されたストリームをスト

50

リームファイルとして記録媒体に記録する記録のステップと、

上記記録媒体に記録される上記ストリームファイルに対し、少なくとも、該ストリームファイルの再生時刻情報とアドレス情報とを対応付けた属性ファイルと、

該ストリームファイルに対して再生開始点と再生終了点とを設定することにより再生区間を指定する１以上の再生区間データが格納され、上記ストリームファイルに対する再生時刻情報を示すマーク情報が格納可能な再生リストファイルと

を生成する管理情報生成のステップと、

上記記録のステップおよび上記管理情報生成のステップとを制御する制御のステップとを有し、

上記制御のステップは、

上記記録指示入力 of のステップによる指示に基づく記録開始から記録停止までの区間に対応する上記ビデオデータおよびオーディオデータを１の上記ストリームファイルとして上記記録媒体に記録するように上記記録のステップを制御すると共に、該１のストリームファイルの全体を再生区間として指定する上記再生区間データを生成し、該再生区間データが指定する上記ストリームファイルの先頭の位置に対応する上記マーク情報を上記再生リストファイルに格納するように上記管理情報生成のステップを制御する記録方法。

【請求項１０】

ビデオデータとオーディオデータとを多重化して記録媒体に記録する記録方法をコンピュータ装置に実行させる記録プログラムにおいて、

上記記録方法は、

データ入力部に入力されたビデオデータおよびオーディオデータの記録開始および記録停止の指示が入力される記録指示入力のステップと、

上記ビデオデータおよびオーディオデータを多重化し、多重化されたストリームをストリームファイルとして記録媒体に記録する記録のステップと、

上記記録媒体に記録される上記ストリームファイルに対し、少なくとも、該ストリームファイルの再生時刻情報とアドレス情報とを対応付けた属性ファイルと、

該ストリームファイルに対して再生開始点と再生終了点とを設定することにより再生区間を指定する１以上の再生区間データが格納され、上記ストリームファイルに対する再生時刻情報を示すマーク情報が格納可能な再生リストファイルと

を生成する管理情報生成のステップと、

上記記録のステップおよび上記管理情報生成のステップとを制御する制御のステップとを有し、

上記制御のステップは、

上記記録指示入力のステップによる指示に基づく記録開始から記録停止までの区間に対応する上記ビデオデータおよびオーディオデータを１の上記ストリームファイルとして上記記録媒体に記録するように上記記録のステップを制御すると共に、該１のストリームファイルの全体を再生区間として指定する上記再生区間データを生成し、該再生区間データが指定する上記ストリームファイルの先頭の位置に対応する上記マーク情報を上記再生リストファイルに格納するように上記管理情報生成のステップを制御する記録プログラム。

【請求項１１】

撮像部で被写体を撮像して得られたビデオデータと、收音部で音声を收音して得られたオーディオデータとを多重化して記録媒体に記録する撮像装置において、

被写体を撮像してビデオデータを出力する撮像部と、

音声を收音してオーディオデータを出力する收音部と、

上記ビデオデータおよび上記オーディオデータを多重化し、多重化されたストリームをストリームファイルとして記録媒体に記録する記録部と、

上記ビデオデータおよび上記オーディオデータの上記記録媒体への記録開始および記録停止を指示するユーザ操作を受け付ける操作部と、

上記記録媒体に記録される上記ストリームファイルに対し、少なくとも、該ストリームファイルの再生時刻情報とアドレス情報とを対応付けた属性ファイルと、

該ストリームファイルに対して再生開始点と再生終了点とを設定することにより再生区間を指定する 1 以上の再生区間データが格納され、上記ストリームファイルに対する再生時刻情報を示すマーク情報が格納可能な再生リストファイルと

を生成する管理情報生成部と、

上記操作部に対するユーザ操作に応じて上記記録部の動作を制御すると共に、上記管理情報生成部を制御する制御部と

を有し、

上記制御部は、

上記操作部に対する操作に応じて、上記記録開始から上記記録停止までの区間に対応する上記ビデオデータおよび上記オーディオデータを 1 の上記ストリームファイルとして上記記録媒体に記録するように上記記録部を制御すると共に、該 1 のストリームファイルの全体を再生区間として指定する上記再生区間データを生成し、該再生区間データが指定する上記ストリームファイルの先頭の位置に対応する上記マーク情報を上記再生リストファイルに格納するように上記管理情報生成部を制御する撮像装置。

10

【請求項 12】

撮像部で被写体を撮像して得られたビデオデータと、收音部で音声を收音して得られたオーディオデータとを多重化して記録媒体に記録する撮像装置の撮像方法において、

撮像部で被写体を撮像して得られたビデオデータと、收音部で音声を收音して得られたオーディオデータとを多重化し、多重化されたストリームをストリームファイルとして記録媒体に記録する記録のステップと、

20

操作部に対する上記ビデオデータおよび上記オーディオデータの上記記録媒体への記録開始および記録停止を指示するユーザ操作を受け付けるステップと、

上記記録媒体に記録される上記ストリームファイルに対し、少なくとも、該ストリームファイルの再生時刻情報とアドレス情報とを対応付けた属性ファイルと、

該ストリームファイルに対して再生開始点と再生終了点とを設定することにより再生区間を指定する 1 以上の再生区間データが格納され、上記ストリームファイルに対する再生時刻情報を示すマーク情報が格納可能な再生リストファイルと

を生成する管理情報生成のステップと、

上記操作部に対するユーザ操作に応じて上記記録のステップの動作を制御すると共に、上記管理情報生成のステップを制御する制御のステップと

30

を有し、

上記制御のステップは、

上記操作部に対する操作に応じて、上記記録開始から上記記録停止までの区間に対応する上記ビデオデータおよび上記オーディオデータを 1 の上記ストリームファイルとして上記記録媒体に記録するように上記記録のステップを制御すると共に、該 1 のストリームファイルの全体を再生区間として指定する上記再生区間データを生成し、該再生区間データが指定する上記ストリームファイルの先頭の位置に対応する上記マーク情報を上記再生リストファイルに格納するように上記管理情報生成のステップを制御する撮像方法。

【発明の詳細な説明】

【技術分野】

40

【0001】

この発明は、ビデオデータとオーディオデータとを多重化したストリームデータを記録媒体に記録するのに適した記録装置、記録方法および記録プログラム、ならびに、撮像装置および撮像方法に関する。

【背景技術】

【0002】

従来から、記録可能で記録再生装置から取り外し可能とされると共に、記録容量が比較的大きく、ビデオデータとオーディオデータとからなる A V (Audio/Video) データを記録するのに適した記録媒体として、4 . 7 G B (Giga Byte) 以上の記録容量を有する D V D (Digital Versatile Disc) が普及している。特許文献 1 には、記録可能なタイプの D V D

50

に対してDVD - Videoフォーマットで記録する撮像装置が記載されている。

【特許文献1】特開2004 - 350251

【発明の開示】

【発明が解決しようとする課題】

【0003】

このような大容量の記録媒体を、例えばビデオカメラ装置のような、記録の開始および停止を頻繁に繰り返すような記録装置に適用した場合に、記録されたAVストリームのユーザによる取り扱い、例えば記録されたAVストリームのタイトル管理や編集が容易な記録フォーマットが求められている。

【0004】

例えば、ビデオデータおよびオーディオデータの連続同期再生、すなわち実時間再生が保障された再生が必要な所定単位に対して再生区間や再生順序を指定することが可能なフォーマットにおいては、記録された所定単位のAVストリームであるAVストリームファイルを、当該AVストリームファイルの属性を示すファイルと、当該所定単位で再生区間などを指定するファイルとにより管理することで、記録媒体上のAVストリームファイルを加工することなく、当該AVストリームファイルの再生区間や再生順序を自由に設定した編集を容易に行うことができる。このようなファイル管理形態において、編集処理が可能な記録フォーマットが求められている。

【0005】

したがって、この発明の目的は、ビデオデータとオーディオデータとが多重化されたAVストリームを、タイトル管理や編集が容易なように記録媒体に記録可能な記録装置、記録方法および記録プログラム、ならびに、撮像装置および撮像方法を提供することにある。

【課題を解決するための手段】

【0006】

上述した課題を解決するために、第1の発明は、ビデオデータとオーディオデータとを多重化して記録媒体に記録する記録装置において、ビデオデータおよびオーディオデータが入力されるデータ入力部と、ビデオデータおよびオーディオデータの記録開始および記録停止の指示が入力される記録指示入力部と、ビデオデータおよびオーディオデータを多重化し、多重化されたストリームをストリームファイルとして記録媒体に記録する記録部と、記録媒体に記録されるストリームファイルに対し、少なくとも、ストリームファイルの再生時刻情報とアドレス情報とを対応付けた属性ファイルと、ストリームファイルに対して再生開始点と再生終了点とを設定することにより再生区間を指定する1以上の再生区間データが格納され、ストリームファイルに対する再生時刻情報を示すマーク情報が格納可能な再生リストファイルとを生成する管理情報生成部と、記録部および管理情報生成部とを制御する制御部とを有し、制御部は、記録指示入力部の指示に基づく記録開始から記録停止までの区間に対応するビデオデータおよびオーディオデータを1のストリームファイルとして記録媒体に記録するように記録部を制御すると共に、1のストリームファイルの全体を再生区間として指定する再生区間データを生成し、再生区間データが指定するストリームファイルの先頭の位置に対応するマーク情報を再生リストファイルに格納するように管理情報生成部を制御する記録装置である。

【0007】

また、第2の発明は、ビデオデータとオーディオデータとを多重化して記録媒体に記録する記録方法において、データ入力部に入力されたビデオデータおよびオーディオデータの記録開始および記録停止の指示が入力される記録指示入力のステップと、ビデオデータおよびオーディオデータを多重化し、多重化されたストリームをストリームファイルとして記録媒体に記録する記録のステップと、記録媒体に記録されるストリームファイルに対し、少なくとも、ストリームファイルの再生時刻情報とアドレス情報とを対応付けた属性ファイルと、ストリームファイルに対して再生開始点と再生終了点とを設定することにより再生区間を指定する1以上の再生区間データが格納され、ストリームファイルに対する

10

20

30

40

50

再生時刻情報を示すマーク情報が格納可能な再生リストファイルとを生成する管理情報生成のステップと、記録のステップおよび管理情報生成のステップとを制御する制御のステップとを有し、制御のステップは、記録指示入力ステップによる指示に基づく記録開始から記録停止までの区間に対応するビデオデータおよびオーディオデータを1のストリームファイルとして記録媒体に記録するように記録のステップを制御すると共に、1のストリームファイルの全体を再生区間として指定する再生区間データを生成し、再生区間データが指定するストリームファイルの先頭の位置に対応するマーク情報を再生リストファイルに格納するように管理情報生成のステップを制御する記録方法である。

【0008】

また、第3の発明は、ビデオデータとオーディオデータとを多重化して記録媒体に記録する記録方法をコンピュータ装置に実行させる記録プログラムにおいて、記録方法は、データ入力部に入力されたビデオデータおよびオーディオデータの記録開始および記録停止の指示が入力される記録指示入力ステップと、ビデオデータおよびオーディオデータを多重化し、多重化されたストリームをストリームファイルとして記録媒体に記録する記録のステップと、記録媒体に記録されるストリームファイルに対し、少なくとも、ストリームファイルの再生時刻情報とアドレス情報とを対応付けた属性ファイルと、ストリームファイルに対して再生開始点と再生終了点とを設定することにより再生区間を指定する1以上の再生区間データが格納され、ストリームファイルに対する再生時刻情報を示すマーク情報が格納可能な再生リストファイルとを生成する管理情報生成のステップと、記録のステップおよび管理情報生成のステップとを制御する制御のステップとを有し、制御のステップは、記録指示入力ステップによる指示に基づく記録開始から記録停止までの区間に対応するビデオデータおよびオーディオデータを1のストリームファイルとして記録媒体に記録するように記録のステップを制御すると共に、1のストリームファイルの全体を再生区間として指定する再生区間データを生成し、再生区間データが指定するストリームファイルの先頭の位置に対応するマーク情報を再生リストファイルに格納するように管理情報生成のステップを制御する記録プログラムである。

【0009】

また、第4の発明は、撮像部で被写体を撮像して得られたビデオデータと、收音部で音声を收音して得られたオーディオデータとを多重化して記録媒体に記録する撮像装置において、被写体を撮像してビデオデータを出力する撮像部と、音声を收音してオーディオデータを出力する收音部と、ビデオデータおよびオーディオデータを多重化し、多重化されたストリームをストリームファイルとして記録媒体に記録する記録部と、ビデオデータおよびオーディオデータの記録媒体への記録開始および記録停止を指示するユーザ操作を受け付ける操作部と、記録媒体に記録されるストリームファイルに対し、少なくとも、ストリームファイルの再生時刻情報とアドレス情報とを対応付けた属性ファイルと、ストリームファイルに対して再生開始点と再生終了点とを設定することにより再生区間を指定する1以上の再生区間データが格納され、ストリームファイルに対する再生時刻情報を示すマーク情報が格納可能な再生リストファイルとを生成する管理情報生成部と、操作部に対するユーザ操作に応じて記録部の動作を制御すると共に、管理情報生成部を制御する制御部とを有し、制御部は、操作部に対する操作に応じて、記録開始から記録停止までの区間に対応するビデオデータおよびオーディオデータを1のストリームファイルとして記録媒体に記録するように記録部を制御すると共に、1のストリームファイルの全体を再生区間として指定する再生区間データを生成し、再生区間データが指定するストリームファイルの先頭の位置に対応するマーク情報を再生リストファイルに格納するように管理情報生成部を制御する撮像装置である。

【0010】

また、第5の発明は、撮像部で被写体を撮像して得られたビデオデータと、收音部で音声を收音して得られたオーディオデータとを多重化して記録媒体に記録する撮像装置の撮像方法において、撮像部で被写体を撮像して得られたビデオデータと、收音部で音声を收音して得られたオーディオデータとを多重化し、多重化されたストリームをストリームフ

10

20

30

40

50

ファイルとして記録媒体に記録する記録のステップと、操作部に対するビデオデータおよびオーディオデータの記録媒体への記録開始および記録停止を指示するユーザ操作を受け付けるステップと、記録媒体に記録されるストリームファイルに対し、少なくとも、ストリームファイルの再生時刻情報とアドレス情報とを対応付けた属性ファイルと、ストリームファイルに対して再生開始点と再生終了点とを設定することにより再生区間を指定する1以上の再生区間データが格納され、ストリームファイルに対する再生時刻情報を示すマーク情報が格納可能な再生リストファイルとを生成する管理情報生成のステップと、操作部に対するユーザ操作に応じて記録のステップの動作を制御すると共に、管理情報生成のステップを制御する制御のステップとを有し、制御のステップは、操作部に対する操作に応じて、記録開始から記録停止までの区間に対応するビデオデータおよびオーディオデータを1のストリームファイルとして記録媒体に記録するように記録のステップを制御すると共に、1のストリームファイルの全体を再生区間として指定する再生区間データを生成し、再生区間データが指定するストリームファイルの先頭の位置に対応するマーク情報を再生リストファイルに格納するように管理情報生成のステップを制御する撮像方法である。

【0012】

上述したように、第1、第2および第3の発明は、データ入力部に入力されたビデオデータおよびオーディオデータを多重化して記録媒体に記録されるストリームファイルに対し、少なくとも、ストリームファイルの再生時刻情報とアドレス情報とを対応付けた属性ファイルと、ストリームファイルに対して再生開始点と再生終了点とを設定することにより再生区間を指定する1以上の再生区間データが格納され、ストリームファイルに対する再生時刻情報を示すマーク情報が格納可能な再生リストファイルとを生成し、記録指示入力による指示に基づく記録開始から記録停止までの区間に対応するビデオデータおよびオーディオデータを1のストリームファイルとして記録媒体に記録するように制御すると共に、1のストリームファイルの全体を再生区間として指定する再生区間データを生成し、再生区間データが指定する再生開始時刻に対応する時刻情報を示すマーク情報を再生リストファイルに格納するように制御するようにしているため、ストリームファイルを、クリップを単位としてプレイリストを用いて管理する管理構造で記録媒体に記録することができる。

【0013】

また、第4、第5および第6の発明は、撮像部で被写体を撮像して得られたビデオデータと、收音部で音声を收音して得られたオーディオデータとを多重化したストリームが記録媒体に記録されるストリームファイルに対し、少なくとも、ストリームファイルの再生時刻情報とアドレス情報とを対応付けた属性ファイルと、ストリームファイルに対して再生開始点と再生終了点とを設定することにより再生区間を指定する1以上の再生区間データが格納され、ストリームファイルに対する再生時刻情報を示すマーク情報が格納可能な再生リストファイルとを生成し、操作部に対する操作に応じて、記録開始から記録停止までの区間に対応するビデオデータおよびオーディオデータを1のストリームファイルとして記録媒体に記録するように制御すると共に、1のストリームファイルの全体を再生区間として指定する再生区間データを生成し、再生区間データが指定する再生開始時刻に対応する時刻情報を示すマーク情報を再生リストファイルに格納するように制御するようにしているため、被写体を撮像して得られたビデオデータおよびオーディオデータを、クリップを単位としてプレイリストを用いて管理する管理構造で記録媒体に記録することができる。

【発明の効果】

【0014】

上述のように、第1、第2および第3の発明は、データ入力部に入力されたビデオデータおよびオーディオデータを多重化して記録媒体に記録されるストリームファイルに対し、少なくとも、ストリームファイルの再生時刻情報とアドレス情報とを対応付けた属性ファイルと、ストリームファイルに対して再生開始点と再生終了点とを設定することにより再生区間を指定する1以上の再生区間データが格納され、ストリームファイルに対する再

生時刻情報を示すマーク情報が格納可能な再生リストファイルとを生成し、記録指示入力による指示に基づく記録開始から記録停止までの区間に対応するビデオデータおよびオーディオデータを1のストリームファイルとして記録媒体に記録するように制御すると共に、1のストリームファイルの全体を再生区間として指定する再生区間データを生成し、再生区間データが指定する再生開始時刻に対応する時刻情報を示すマーク情報を再生リストファイルに格納するように制御するようにしているため、ストリームファイルを、クリップを単位としてプレイリストを用いて管理する管理構造で記録媒体に記録することができる効果がある。

【0015】

また、第4、第5および第6の発明は、撮像部で被写体を撮像して得られたビデオデータと、收音部で音声を收音して得られたオーディオデータとを多重化したストリームが記録媒体に記録されるストリームファイルに対し、少なくとも、ストリームファイルの再生時刻情報とアドレス情報とを対応付けた属性ファイルと、ストリームファイルに対して再生開始点と再生終了点とを設定することにより再生区間を指定する1以上の再生区間データが格納され、ストリームファイルに対する再生時刻情報を示すマーク情報が格納可能な再生リストファイルとを生成し、操作部に対する操作に応じて、記録開始から記録停止までの区間に対応するビデオデータおよびオーディオデータを1のストリームファイルとして記録媒体に記録するように制御すると共に、1のストリームファイルの全体を再生区間として指定する再生区間データを生成し、再生区間データが指定する再生開始時刻に対応する時刻情報を示すマーク情報を再生リストファイルに格納するように制御するようにしているため、被写体を撮像して得られたビデオデータおよびオーディオデータを、クリップを単位としてプレイリストを用いて管理する管理構造で記録媒体に記録することができる効果がある。

【発明を実施するための最良の形態】

【0016】

以下、この発明の実施の一形態を、図面を参照しながら説明する。先ず、理解を容易とするために、この発明に適用可能な一例のフォーマット（以下、AVCHDフォーマットと呼ぶ）について説明する。AVCHDフォーマットは、ビデオデータとオーディオデータとが所定に多重化されたAV(Audio/Video)ストリームを記録可能な記録媒体に記録する記録フォーマットとして現在提案されているもので、記録媒体に記録されたAVストリームを、クリップ単位でプレイリストを用いて管理可能としている。

【0017】

例えばITU-T(International Telecommunication Union-Telecommunication Standardization Sector)勧告H.264あるいはISO(International Organization for Standardization)/IEC(International Electrotechnical Commission)国際標準14496-10(MPEG-4パート10)Advanced Video Coding(以下、H.264|AVCと略称する)に規定される符号化方式や、MPEG(Moving Pictures Experts Group)ビデオやMPEGオーディオといった符号化方式で符号化され、MPEG2システムズに従い多重化されたビットストリームは、クリップAVストリーム(またはAVストリーム)と称される。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ディスクに記録される。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。

【0018】

クリップAVストリームファイルは、ファイルシステム上での管理単位であり、ユーザにとって必ずしも分かりやすい管理単位であるとは限らない。ユーザの利便性を考えた場合、複数のクリップAVストリームファイルに分割された映像コンテンツを一つにまとめて再生する仕組みや、クリップAVストリームファイルの一部だけを再生する仕組み、さらには、特殊再生や頭出し再生を滑らかに行うための情報などをデータベースとしてディスクに記録しておく必要がある。

【0019】

図1は、この発明に適用可能なAVCHDフォーマットに規定されるデータモデルを概略的に示す。このAVCHDフォーマットによれば、データ構造は、図1に示されるように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ストリームに対する再生箇所を指定するための、プレイリスト(Playlist)と、プレイアイテム(PlayItem)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、プレイリスト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(Movie Object)など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、記録媒体に格納されるタイトルなどを管理するインデックステーブルが配置される（便宜上、インデックスレイヤと呼ぶ）。

10

【0020】

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。このクリップAVストリームに関する情報がクリップ情報(Clip Information)としてファイルに記録される。

【0021】

また、クリップAVストリームには、字幕を表示するグラフィクスストリームであるOBストリーム(Overlay Bitmap stream)や、メニュー表示などに用いられるデータ（ボタン画像データなど）をストリームにしたMBストリーム(Menu Bitmap stream)ストリームを多重化することができる。

20

【0022】

クリップAVストリームファイルと、対応するクリップ情報が記録されたクリップ情報ファイルと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。すなわち、クリップは、クリップAVストリームとクリップ情報とから構成される、一つのオブジェクトである。

【0023】

ファイルは、一般的に、バイト列として扱われる。クリップAVストリームファイルのコンテンツは、時間軸上に展開され、クリップ中のエン트리ポイントは、主に時間ベースで指定される。所定のクリップへのアクセスポイントのタイムスタンプが与えられた場合、クリップAVストリームファイルの中でデータの読み出しを開始すべきアドレス情報を見つけるために、クリップ情報ファイルを用いることができる。

30

【0024】

プレイリストレイヤについて説明する。プレイリストは、再生するAVストリームファイルの指定と、指定されたAVストリームファイル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(PlayItem)と称される。プレイリストは、プレイアイテムの集合で構成される。プレイアイテムを再生するということは、そのプレイアイテムに参照されるAVストリームファイルの一部分を再生するということになる。すなわち、プレイアイテム中のIN点およびOUT点情報に基づき、クリップ中の対応する区間が再生される。

40

【0025】

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は、ナビゲーションコマンドプログラムと、ムービーオブジェクトとを連携するターミナルインフォメーションを含む。ナビゲーションプログラムは、プレイリストの再生を制御するためのコマンド（ナビゲーションコマンド:navigation command）である。ターミナルインフォメーションは、ユーザのプレーヤに対するインタラクティブな操作を許可するための情報を含んでいる。このターミナルインフォメーションに基づき、メニュー画面の呼び出しや、タイトルサーチといったユーザオペレーションが制御される。

【0026】

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

50

(Index Table)からなる。インデックステーブルは、記録媒体に記録されたコンテンツのタイトルを定義する、トップレベルのテーブルである。インデックステーブルに格納されているタイトル情報に基づき、プレーヤに常駐されるシステムソフトウェア中のモジュールマネージャにより記録媒体の再生が制御される。

【 0 0 2 7 】

すなわち、図 2 に概略的に示されるように、インデックステーブル中の任意のエントリは、タイトルと称され、インデックステーブルにエントリされるファーストプレイバックタイトル(First PlaybackTitle)、メニュータイトル(MenuTitle)およびムービータイトル(MovieTitle) # 1、# 2、・・・は、全てタイトルである。各タイトルは、ムービーオブジェクトに対するリンクを示す。

10

【 0 0 2 8 】

理解を容易とするため再生専用の記録媒体を例にとると、例えば、ファーストプレイバックタイトルは、当該記録媒体に格納されるコンテンツが映画であれば、映画本編に先立って映出される映画会社の宣伝用映像(トレーラ)に対応する。メニュータイトルは、例えばコンテンツが映画である場合、本編再生、チャプタサーチ、字幕や言語設定、特典映像再生などを選択するためのメニュー画面に対応する。また、ムービータイトルは、メニュータイトルから選択される各映像である。タイトルがさらにメニュー画面であるような構成も可能である。

【 0 0 2 9 】

図 3 は、上述のようなクリップ A V ストリーム、クリップ情報(Stream Attributes)、クリップ、プレイアイテムおよびプレイリストの関係を示す U M L (Unified Modeling Language) 図である。プレイリストは、1 または複数のプレイアイテムに対応付けられ、プレイアイテムは、1 のクリップに対応付けられる。1 のクリップに対して、それぞれ開始点および / または終了点が異なる複数のプレイアイテムを対応付けることができる。1 のクリップから 1 のクリップ A V ストリームファイルが参照される。同様に、1 のクリップから 1 のクリップ情報ファイルが参照される。また、クリップ A V ストリームファイルとクリップ情報ファイルとは、1 対 1 の対応関係を有する。このような構造を定義することにより、クリップ A V ストリームファイルを変更することなく、任意の部分だけを再生する、非破壊の再生順序指定を行うことが可能となる。

20

【 0 0 3 0 】

また、図 4 のように、複数のプレイリストから同一のクリップを参照することもできる。また、1 のプレイリストから複数のクリップを指定することもできる。クリップは、プレイリスト中のプレイアイテムに示される I N 点および O U T 点により、参照される。図 4 の例では、クリップ 3 0 0 は、プレイリスト 3 1 0 のプレイアイテム 3 2 0 から参照されると共に、プレイリスト 3 1 1 を構成するプレイアイテム 3 2 1 および 3 2 2 のうちプレイアイテム 3 2 1 から、I N 点および O U T 点で示される区間が参照される。また、クリップ 3 0 1 は、プレイリスト 3 1 1 のプレイアイテム 3 2 2 から I N 点および O U T 点で示される区間が参照されると共に、プレイリスト 3 1 2 のプレイアイテム 3 2 3 および 3 2 4 のうち、プレイアイテム 3 2 3 の I N 点および O U T 点で示される区間が参照される。

30

40

【 0 0 3 1 】

次に、A V C H D フォーマットによる、記録媒体に記録されるファイルの管理構造について、図 5 を用いて説明する。ファイルは、ディレクトリ構造により階層的に管理される。記録媒体上には、まず、1 つのディレクトリ(図 5 の例ではルート(root)ディレクトリ)が作成される。このディレクトリの下が、1 つの記録再生システムで管理される範囲とする。

【 0 0 3 2 】

ルートディレクトリの下に、ディレクトリ"BDMV"およびディレクトリ"AVCHD TN"が置かれる。ディレクトリ"AVCHD TN"には、例えばクリップの代表画像を所定サイズに縮小したサムネイルファイルが置かれる。ディレクトリ"BDMV"に、図 1 を用いて説明したデータ構

50

造が格納される。

【 0 0 3 3 】

ディレクトリ"BDMV"の直下には、ファイルは、ファイル"index.bdmv"およびファイル"MovieObject.bdmv"の2つのみを置くことができる。また、ディレクトリ"BDMV"の下に、ディレクトリ"PLAYLIST"、ディレクトリ"CLIPINF"、ディレクトリ"STREAM"およびディレクトリ"BACKUP"が置かれる。ディレクトリ"BACKUP"は、各ディレクトリおよびファイルのバックアップが格納される。

【 0 0 3 4 】

ファイル"index.bdmv"は、ディレクトリ"BDMV"の内容について記述される。すなわち、このファイル"index.bdmv"が上述した最上層のレイヤであるインデックスレイヤにおけるインデックステーブルに対応する。また、ファイル"MovieObject.bdmv"は、1つ以上のムービーオブジェクトの情報が格納される。すなわち、このファイル"MovieObject.bdmv"が上述したオブジェクトレイヤに対応する。

【 0 0 3 5 】

ディレクトリ"PLAYLIST"は、プレイリストのデータベースが置かれるディレクトリである。すなわち、ディレクトリ"PLAYLIST"は、プレイリストに関するファイルであるファイル"xxxxx.mpls"を含む。ファイル"xxxxx.mpls"は、プレイリストのそれぞれに対して作成されるファイルである。ファイル名において、"."（ピリオド）の前の"xxxxx"は、5桁の数字とされ、ピリオドの後ろの"mpls"は、このタイプのファイルに固定的とされた拡張子である。

【 0 0 3 6 】

ディレクトリ"CLIPINF"は、クリップのデータベースが置かれるディレクトリである。すなわち、ディレクトリ"CLIPINF"は、クリップA Vストリームファイルのそれぞれに対するクリップインフォメーションファイルであるファイル"zzzzz.clpi"を含む。ファイル名において、"."（ピリオド）の前の"zzzzz"は、5桁の数字とされ、ピリオドの後ろの"clpi"は、このタイプのファイルに固定的とされた拡張子である。

【 0 0 3 7 】

ディレクトリ"STREAM"は、実体としてのA Vストリームファイルが置かれるディレクトリである。すなわち、ディレクトリ"STREAM"は、クリップインフォメーションファイルのそれぞれに対応するクリップA Vストリームファイルを含む。クリップA Vストリームファイルは、MPEG2 (Moving Pictures Experts Group 2)のトランスポートストリーム（以下、MPEG2 TSと略称する）からなり、ファイル名が"zzzzz.m2ts"とされる。ファイル名において、ピリオドの前の"zzzzz"は、対応するクリップインフォメーションファイルと同一することで、クリップインフォメーションファイルとこのクリップA Vストリームファイルとの対応関係を容易に把握することができる。

【 0 0 3 8 】

なお、ディレクトリ"AVCHDTN"は、2種類のサムネイルファイル"thumbnail.tidx"および"thumbnail.tdt2"を置くことができる。サムネイルファイル"thumbnail.tidx"は、所定の方式で暗号化されたサムネイル画像が格納される。サムネイルファイル"thumbnail.tdt2"は、暗号化されていないサムネイル画像が格納される。例えばビデオカメラでユーザが撮影したクリップに対応するサムネイル画像は、コピーフリーであって暗号化する必要が無いと考えられるため、このサムネイルファイル"thumbnail.tdt2"に格納される。

【 0 0 3 9 】

図5で示した各ファイルのうち、この発明に関わりの深いものについて、より詳細に説明する。先ず、ディレクトリ"BDMV"の直下に置かれるファイル"index.bdmv"について説明する。図6は、このファイル"index.bdmv"の一例の構造を表すシンタクスを示す。ここでは、シンタクスをコンピュータ装置などのプログラムの記述言語として用いられるC言語の記述法に基づき示す。これは、他のシンタクスを表す図において、同様である。

【 0 0 4 0 】

図6において、フィールドTypeIndicatorは、32ビットのデータ長を有し、このファ

10

20

30

40

50

イルがインデックステーブルであることを示す。フィールドTypeIndicator2は、32ビットのデータ長を有し、このファイル"index.bdmv"のバージョンを示す。フィールドIndexesStartAddressは、32ビットのデータ長を有し、このシンタクス内にあるブロックblkIndexes()の開始アドレスを示す。

【0041】

フィールドExtensionDataStartAddressは、32ビットのデータ長を有し、このシンタクス内にあるブロックblkExtensionData()の開始アドレスを示す。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。フィールドExtensionDataStartAddressは、このファイル"index.bdmv"の最初のバイトからの相対バイト数で、ブロックblkExtensionData()の開始アドレスを示す。相対バイト数は、"0"から開始される。若し、このフィールドExtensionDataStartAddressの値が"0"であれば、このファイル"index.bdmv"内に、ブロックblkExtensionData()が存在しないことを示す。

【0042】

フィールドExtensionDataStartAddressに続けて、データ長が192バイトの領域reservedが配される。なお、領域reservedは、バイトアライメントや、将来的なフィールドの追加などのための領域である。これは、以下の説明においても同様である。ブロックblkAppInfoBDMV()は、コンテンツ制作者が任意の情報を記述できるブロックであって、プレーヤの動作などには影響を与えない。

【0043】

ブロックblkIndexes()は、このファイル"index.bdmv"の実質的な内容であって、このブロックblkIndexes()に記述された内容により、ディスクをプレーヤに装填した際に再生されるファーストプレイバックや、トップメニューから呼び出されるタイトル(ムービーオブジェクト)が指定される。インデックステーブルにより呼び出されたムービーオブジェクト等に記述されたコマンドに基づき、後述するプレイリストファイルが読み込まれる。

【0044】

図7は、ブロックblkIndexes()の一例の構造を表すシンタクスを示す。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blkIndexes()の終わりまでのデータ長を示す。続けて、ブロックFirstPlaybackTitle()およびブロックMenuTitle()が配される。

【0045】

ブロックFirstPlaybackTitle()は、ファーストプレイバックで用いられるオブジェクトに関する情報が記述される。ブロックFirstPlaybackTitle()は、1ビットのデータ長を有する領域reservedに続けて固定値"1"が記述される。さらに31ビットのデータ長を有する領域reservedを介して固定値"1"が記述される。そして、14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FirstPlaybackTitleMobjIDRefが配される。このフィールドFirstPlaybackTitleMobjIDRefにより、ファーストプレイバックタイトルで用いられるムービーオブジェクトのIDを示す。

【0046】

ムービーオブジェクトのIDは、例えば、図8および図9を用いて後述するムービーオブジェクトのシンタクスに基づき、ムービーオブジェクトのforループ文においてループ変数として用いられる値mobj_idで示される。この例では、フィールドFirstPlaybackTitleMobjIDRefは、参照するムービーオブジェクトに対応する値mobj_idが格納される。

【0047】

なお、ブロックblkIndexes()におけるブロックFirstPlaybackTitle()内のフィールドFirstPlaybackTitleMobjIDRefは、トップメニューのムービーオブジェクトを指していてもよいし、タイトルを指していてもよい。

【0048】

ブロックMenuTitle()は、トップメニューで用いられるオブジェクトに関する情報が記述される。ブロックMenuTitle()は、1ビットのデータ長を有する領域reservedに続けて固定値"1"が記述される。さらに31ビットのデータ長を有する領域reservedを介して固

10

20

30

40

50

定値"1"が記述される。そして、14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MenuTitleMobjIDRefが配される。フィールドMenuTitleMobjIDRefは、メニュータイトルで用いられるムービーオブジェクトのIDを示す。

【0049】

ブロックMenuTitle()の次のフィールドNumberOfTitlesは、16ビットのデータ長を有し、ユーザが選択、再生可能なタイトルの数を示す。次のforループ文に従い、このフィールドNumberOfTitlesに示される回数だけ、値title_idを引数として、ブロックMovieTitle[title_id]()が記述される。ブロックMovieTitle[title_id]()は、タイトル毎の情報が記述される。値title_idは、"0"からフィールドNumberOfTitlesで示される値までの数値であり、タイトルを識別する。

10

【0050】

ブロックMovieTitle[title_id]()において、1ビットのデータ長を有する領域reservedを介して固定値"1"が記述され、さらに、46ビットのデータ長を有する領域reservedを介してフィールドMovieTitleMobjIDRefが記述される。フィールドMovieTitleMobjIDRefは、16ビットのデータ長を有し、このタイトルで用いられるムービーオブジェクトのIDを示す。フィールドMovieTitleMobjIDRefの後ろに、32ビットのデータ長を有する領域reservedが配される。

【0051】

図8は、ディレクトリ"BDMV"の直下に置かれるファイル"MovieObject.bdmv"の一例の構造を表すシンタクスを示す。フィールドTypeIndicatorは、32ビット(4バイト)のデータ長を有し、このファイルがファイル"MovieObject.bdmv"であることを示す。フィールドTypeIndicatorは、ISO(International Organization for Standardization)646に規定された符号化方式で符号化した4文字からなる文字列が記述される。この図8の例では、フィールドtype_indicatorにISO646に既定の方式で符号化された4文字の文字列"MOBJ"が記述され、このファイルがファイル"MovieObject.bdmv"であることが示される。

20

【0052】

フィールドTypeIndicator2は、32ビット(4バイト)のデータ長を有し、このファイル"MovieObject.bdmv"のバージョン番号を示す。このファイル"MovieObject.bdmv"では、フィールドTypeIndicator2は、ISO646に規定された符号化方式で符号化した4文字の文字列"0100"でなければならない。

30

【0053】

フィールドExtensionDataStartAddressは、32ビットのデータ長を有し、このシンタクス内にあるブロックblkExtensionData()の開始アドレスを示す。フィールドExtensionDataStartAddressは、このファイル"MovieObject.bdmv"の最初のバイトからの相対バイト数で、ブロックblkExtensionData()の開始アドレスを示す。相対バイト数は、"0"から開始される。若し、このフィールドExtensionDataStartAddressの値が"0"であれば、このファイル"MovieObject.bdmv"内に、ブロックblkExtensionData()が存在しないことを示す。

40

【0054】

なお、この図8に示すシンタクス内のフィールドpadding_wordは、16ビットのデータ長を有し、このファイル"MovieObject.bdmv"のシンタクスに従いforループ文に値N1または値N2で示される回数だけ挿入される。値N1または値N2は、0または任意の正の整数である。また、フィールドpadding_wordは、任意の値を用いることができる。

【0055】

フィールドExtensionDataStartAddressに続けてデータ長が224ビットの領域reservedが配され、その次に、このファイル"MovieObject.bdmv"の本体であるブロックblkMovieObjects()が格納される。

【0056】

50

図 9 は、ブロックblkMovieObjects()の一例の構造を表すシンタクスを示す。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の直後からこのブロックblkMovieObjects()の終わりまでのデータ長を示す。32ビットのデータ長を有する領域reservedを介してフィールドNumberOfMobjsが配される。フィールドNumberOfMobjsは、直後のforループ文に従い格納されるムービーオブジェクトの数を示す。forループ文のループ変数として用いられる値mobj_idで、ムービーオブジェクトが一意に特定される。値mobj_idは、"0"から始まる値で、ムービーオブジェクトは、forループ文中に記述される順序により定義される。

【0057】

forループ文中のブロックTerminalInfo()は、固定値"1"が記述され、次に15ビットのデータ長を有する領域reservedが配される。その次に、16ビットのデータ長を有するフィールドNumberOfNavigationCommands[mobj_id]が配される。このフィールドNumberOfNavigationCommands[mobj_id]は、値mobj_idによって指し示されるムービーオブジェクトMovieObject[mobj_id]()に含まれるナビゲーションコマンド(NavigationCommand)の数を表す。

10

【0058】

次の、値command_idをループ変数とするforループ文により、フィールドNumberOfNavigationCommands[mobj_id]に示される数だけ、ナビゲーションコマンドが記述される。すなわち、このforループ文中に配されるフィールドNavigationCommand[mobj_id][command_id]は、値mobj_idによって指し示されるブロックMovieObject[mobj_id]()に含まれる、値command_idで示される順番のナビゲーションコマンドNavigationCommandを格納する。値command_idは、0から始まる値で、ナビゲーションコマンドNavigationCommandは、このforループ文中に記述される順序で定義される。

20

【0059】

図10は、プレイリストファイル"xxxxx.mpls"の一例の構造を表すシンタクスを示す。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。

30

【0060】

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、ファイル"xxxxx.mpls"の最初のバイトからの相対バイト数を表した値である。相対バイト数は、"0"から開始される。若し、このフィールドExtensionDataStartAddressの値が"0"であれば、このファイル"xxxxx.mpls"内に、ブロックblkExtensionData()が存在しないことを示す。

【0061】

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。

40

【0062】

なお、この図10に示すシンタクス内のフィールドpadding_wordは、16ビットのデータ長を有し、このファイル"xxxxx.mpls"のシンタクスに従いforループ文に値N1、値N2および値N3で示される回数だけ挿入される。値N1、値N2または値N3は、0または任意の正の

50

整数である。また、フィールドpadding_wordは、任意の値を用いることができる。

【 0 0 6 3 】

図 1 1 は、ブロックblkPlayList()の一例の構造を表すシンタクスを示す。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。

【 0 0 6 4 】

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。さらに次のforループ文に従い、フィールドNumberOfSubPathsで示される数だけ、ブロックblkSubPath()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blkSubPath()の識別子SubPath_idとなる。

【 0 0 6 5 】

なお、サブパスは、主として再生されるプレイアイテムに対応するメインパスに対して、サブプレイアイテムに対応して持つことができる。サブパスは、例えば、アフレコ用のオーディオデータの指定や、2枚の映像を合成する際に、プレイアイテムで指定されるクリップと同期して再生する副映像を指定するといった目的で用いられる。

【 0 0 6 6 】

図 1 2 は、ブロックblkPlayItem()の一例の構造を表すシンタクスを示す。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()の最後までまでのデータ長を示す。

【 0 0 6 7 】

フィールドClipInformationFileName[0]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[0]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ClipCodecIdentifier[0]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ストリームのコーデック方式を示す。

【 0 0 6 8 】

12ビットのデータ長を有する領域reservedを介して、フィールドConnectionConditionが配される。フィールドConnectionConditionは、4ビットのデータ長を有し、クリップ間の接続状態に関する情報を示す。記録用途の記録媒体に対しては、フィールドConnectionConditionの値として"1"、"5"または"6"が用いられる。フィールドConnectionConditionの値が"1"で、そのクリップと次のクリップとがシームレス接続しないことを示し、フィールドConnectionConditionの値が"5"または"6"で、そのクリップと次のクリップとがシームレス接続することを示す。なお、シームレス接続とは、クリップと次のクリップとがフレームタイミングで連続的に再生されるように、クリップ間の再生制御を行うことをいう。

【 0 0 6 9 】

フィールドConnectionConditionの値が"5"で、当該プレイアイテムが参照するクリップにおいて、オーディオデータの記録長がビデオデータの記録長に対して長くされる(図 1 3 A 参照)。これにより、クリップとクリップとを接続する際に、オーディオデータのフェードアウト処理が可能とされる。例えば、ユーザによる記録停止操作によりクリップがクローズされる場合に、フィールドConnectionConditionの値が"5"とされる。以下、このフィールドConnectionConditionの値が"5"で示されるクリップの接続方法を、第1のシームレス接続と呼ぶ。

10

20

30

40

50

【 0 0 7 0 】

フィールドConnectionConditionの値が" 6 "で、当該プレイアイテムが参照するクリップにおいて、オーディオデータの記録長がビデオデータの記録長に対して同じか若しくは短いくされる（図 1 3 B 参照）。これにより、クリップとクリップとの接続をシームレスに行うことが可能とされる。例えば、ユーザ操作に応じた記録停止以外の理由、例えばシステム要因に基づきクリップがクローズされる場合に、フィールドConnectionConditionの値が" 6 "とされる。以下、このフィールドConnectionConditionの値が" 6 "で示されるクリップの接続方法を、第 2 のシームレス接続と呼ぶ。

【 0 0 7 1 】

フィールドRefToSTCID[0]は、8 ビットのデータ長を有し、システムタイムベース（S T C）の不連続点に関する情報を示す。フィールドINTimeおよびフィールドOUTTimeは、それぞれ32 ビットのデータ長を有し、メインクリップA Vストリームの再生範囲を示す。フィールドINTimeが開始点（I N点）を示し、フィールドOUTTimeが終了点（O U T点）を示す。

【 0 0 7 2 】

ブロックblkUOMaskTable()は、ユーザ入力の受付制限が設定されるテーブルである。1 ビットのデータ長を有するフラグPlayItemRandomAccessFlagは、このブロックblkPlayItem()によるプレイアイテムに対してランダムアクセスを許可するか否かを規定する。続けて、7 ビットのデータ長を有する領域reservedを介してフィールドStillModeが配される。フィールドStillModeは、8 ビットのデータ長を有し、ブロックblkPlayItem()によるプレイアイテムにおいて、最後に表示した映像を静止画として表示させるか否かを示す。フィールドStillModeの値が"0x01"（バイナリ）であれば、if文に基づき、16 ビットのデータ長を有するフィールドStillTimeにより静止時間が示される。フィールドStillModeの値が"0x01"以外であれば、当該16 ビットのデータ長を有する領域がワードアライメントのために予約された領域reservedとされる。

【 0 0 7 3 】

ブロックblkSTNTable()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A Vストリームの属性、P I D 番号、記録媒体上での記録位置などが管理される。

【 0 0 7 4 】

図 1 4 は、ブロックblkPlayListMark()の一例の構造を表すシンタクスを示す。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()の最後までデータの長を示す。

【 0 0 7 5 】

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。

【 0 0 7 6 】

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。フィールドRefToPlayItemID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PlayItem_idが記述される。フィールドMarkTimeStamp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EntryESPIDは、16 ビットのデータ長を有し、マークによって指し示されるエレメンタリストリームを含んでいるT S パケットのP I D の値を示す。フィールドDurationは、45 k H z 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。

【 0 0 7 7 】

10

20

30

40

50

図 15 は、クリップインフォメーションファイルの一例の構造を表すシンタクスを示す。フィールドTypeIndicatorは、32ビット(4バイト)のデータ長を有し、このファイルがクリップインフォメーション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、このクリップインフォメーションファイルのバージョンを示す。

【0078】

このクリップインフォメーションファイルは、ブロックblkClipInfo()、ブロックblkSequenceInfo()、ブロックblkProgramInfo()、ブロックblkCPI()、ブロックblkClipMark()およびブロックblkExtensionData()を有し、それぞれ32ビットのデータ長を有するフィールドSequenceInfoStartAddress、フィールドProgramInfoStartAddress、フィールドCPIStartAddress、フィールドClipMarkStartAddressおよびフィールドExtensionDataStartAddressは、各々対応するブロックの開始アドレスを示す。

【0079】

フィールドExtensionDataStartAddressは、このクリップインフォメーションファイルの最初のバイトからの相対バイト数で、ブロックblkExtensionData()の開始アドレスを示す。相対バイト数は、"0"から開始される。若し、このフィールドExtensionDataStartAddressの値が"0"であれば、このファイル"index.bdmv"内に、ブロックblkExtensionData()が存在しないことを示す。

【0080】

ブロックblkClipInfo()は、これらの開始アドレスを示すフィールドに続く、96ビットのデータ長を有する領域reservedの次から開始される。ブロックblkClipInfo()は、このクリップインフォメーションファイルが管理するクリップAVストリームに関する情報が記述される。ブロックblkSequenceInfo()は、STCやATC(アライバルタイムベース)が連続しているシーケンスをまとまりとして管理する情報が記述される。ブロックblkProgramInfo()は、このクリップインフォメーションファイルに管理されるクリップAVストリームの符号化方式、クリップAVストリーム中のビデオデータのアスペクト比などの情報が記述される。ブロックblkCPI()は、ランダムアクセス開始点などの、AVストリーム中の特徴的な箇所を表す特徴点情報CPIに関する情報が格納される。

【0081】

また、ブロックblkClipMark()は、チャプタ位置などの、クリップに付された頭出しのためのインデックス点(ジャンプポイント)が記述される。ブロックblkExtensionData()は、拡張データを格納することができる領域である。なお、これらブロックblkClipMark()およびクリップインフォメーションファイル内のブロックblkExtensionData()は、この発明との関連性が薄いので、詳細な説明を省略する。

【0082】

図 16 は、ブロックblkClipInfo()の一例の構造を表すシンタクスを示す。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blkClipInfo()の最後までまでのデータ長を示す。16ビットのデータ長を有する領域reservedを介して、フィールドClipStreamTypeが配される。

【0083】

フィールドClipStreamTypeは、8ビットのデータ長を有し、クリップAVストリームの種別を表す。このフィールドClipStreamTypeの値は、例えば"1"に固定的とされる。フィールドApplicationTypeは、8ビットのデータ長を有し、クリップAVストリーム(拡張子が「m2ts」のファイル)がどのような多重化によって作られているかを示す。フィールドApplicationTypeの値が"1"で、対応するクリップAVストリームは、通常の動画が再生される。続けて31ビットのデータ長を有する領域reservedが配される。

【0084】

データ長が1ビットのフラグIsCC5は、プレイリストにおけるブロックblkPlayItem()によって、対応するクリップと次のクリップとの接続を、上述した第1のシームレス接続、すなわちフィールドConnectionConditionの値が"5"で示される方法で行うか否かを示す

10

20

30

40

50

。フラグIsCC5の値が"1" (バイナリ値) であれば、クリップ間の接続が第1のシームレス接続によりなされていることを示す。

【0085】

フィールドTSRecordingRateは、クリップAVストリームファイルの記録レートをバイト/秒で表したものである。フィールドNumberOfSourcePacketsは、クリップAVストリームに含まれるパケット数を表す。システムにより予約された1024ビットのデータ長の領域reservedを介してブロックTSTypeInfoBlock()が配される。ブロックTSTypeInfoBlock()は、クリップAVストリームが格納されるパケットのタイプを示す情報が格納される。このブロックTSTypeInfoBlock()は、この発明との関連性が薄いので、詳細な説明を省略する。

10

【0086】

次のif文以下の情報は、上述のフラグIsCC5の値が"1"、すなわち、対応するクリップと次のクリップとの接続を第1のシームレス接続で行うとされた場合に記述される。if文の次の8ビットのデータ長を有する領域reservedを介してフィールドFollowingClipStreamTypeが配されるフィールドFollowingClipStreamTypeは、8ビットのデータ長を有し、このクリップインフォメーションファイルに対応するクリップの次のクリップのタイプが記述される。8ビットのデータ長を有する領域reservedを介してフィールドFollowingClipInformationFileNameが配される。

【0087】

フィールドFollowingClipInformationFileNameは、40ビット(5バイト)のデータ長を有し、このクリップインフォメーションファイルに対応するクリップの次のクリップに対応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記述される。次のフィールドClipCodecIdentifierは、32ビット(4バイト)のデータ長を有し、当該次のクリップの符号化方式を示す。この例では、フィールドClipCodecIdentifierは、ISO646に既定の方式で符号化された4文字の文字列値"M2TS"に固定的とされる。次に8ビットのデータ長を有する領域reservedが配される。

20

【0088】

図17は、ブロックblkSequenceInfo()の一例の構造を表すシンタクスを示す。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blkSequenceInfo()の最後までのデータ長を示す。15ビットのデータ長を有する領域reservedを介してデータ長が1ビットで固定値"1"が記述される。

30

【0089】

次のフィールドSPNATCStartは、32ビットのデータ長を有し、連続した時間に記録されたことを表すシーケンス(シーケンスATCSequenceと呼ぶ)の開始をパケット番号で表す。この図17の例では、フィールドSPNATCStartは、値を"0"としてクリップAVストリームファイルの先頭と一致させている。フィールドNumberOfSTCSequenceは、シーケンスATCSequence上のシーケンスSTCSequenceの数を表す。フィールドNumberOfSTCSequenceは、値が"1"以上とされる。

【0090】

次のforループ文に従い、フィールドNumberOfSTCSequenceで示される数だけ、シーケンスSTCSequenceの情報が記述される。シーケンスSTCSequenceは、MPEG2 TS (Transport Stream)における時間軸の基準であるPCR (Program Clock Reference)が連続な範囲を表す。シーケンスSTCSequenceには、クリップ内で一意な番号STC_idが割り当てられる。このシーケンスSTCSequence内では、不連続の無い一貫した時間軸を定義できるので、プレイアイテムの開始時刻および終了時刻を一意に定めることができる。つまり、各プレイアイテムの開始点と終了点は、同一のシーケンスSTCSequenceに存在していなければならない。このforループ文においては、値stc_idによりシーケンスSTCSequenceが指定される。

40

【0091】

フィールドPCRPID[stc_id]は、16ビットのデータ長を有し、MPEG2 TSにおい

50

て、PCR (Program Clock Reference)が含まれるTSパケットのPIDを表す。フィールドSPNSTCStart[stc_id]は、32ビットのデータ長を有し、シーケンスSTCSequenceの開始をパケット番号で表す。フィールドPresentationStartTimeおよびフィールドPresentationEndTimeは、それぞれ32ビットのデータ長を有し、クリップAVストリーム中の有効な範囲を表す。フィールドPresentationStartTimeおよびフィールドPresentationEndTimeで示される範囲がプレイアイテムから参照できる範囲となる。

【0092】

図18は、ブロックblkProgramInfo()の一例の構造を表すシンタクスを示す。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blkProgramInfo()の最後までデータ長を示す。15ビットのデータ長を有する領域reservedを介して、データ長が1ビットで固定値"1"が記述される。

【0093】

フィールドSPNProgramSequenceStartは、32ビットのデータ長を有し、対応するクリップAVストリームファイルにおいて、プログラムシーケンスが開始されるソースパケットの番号が記述される。フィールドProgramMapPIDは、16ビットのデータ長を有し、プログラムシーケンスに適用可能なプログラムマップセクションを含むとされているTSパケットのPIDの値を示す。フィールドNumberOfStreamsInPSは、8ビットのデータ長を有し、プログラムシーケンスに定義されるエレメンタリストリームの数を示す。フィールドNumberOfStreamsInPSに続けて、8ビットのデータ長を有する領域reservedが配される。

【0094】

次のforループ文に従い、値[stream_index]をループ変数として、フィールドNumberOfStreamsInPSで示される数だけ、フィールドStreamPID[stream_index]およびブロックblkStreamCodingInfo(stream_index)の組が格納される。フィールドStreamPID[stream_index]は、プログラムシーケンスによって参照されたPMT (Program Map Table)に記述されたエレメンタリストリームに対応するPIDの値を示す。次のブロックblkStreamCodingInfo(stream_index)は、対応するフィールドStreamPID[stream_index]で示されるエレメンタリストリームの符号化方式に関する情報が記述される。

【0095】

図19は、ブロックblkCPI()の一例の構造を表すシンタクスを示す。MPEGストリームのような、フレーム間圧縮を行っている符号化ストリームにおいては、デコード開始可能な箇所は、GOP (Group Of Picture)の先頭など一部の箇所に限られていることが多い。CPI (Characteristic Point Information)とは、そのデコード可能な開始点の位置の情報を集めたデータベースで、再生時刻と、ファイル内アドレスとが対応付けられたテーブルになっている。すなわち、CPIは、デコード単位の先頭位置を示す情報がテーブル化されている。

【0096】

このようにデータベースを定めることで、例えば、任意の時刻から再生したい場合、再生時刻を元にCPIを参照することによって再生位置のファイル内アドレスがわかる。このアドレスは、デコード単位の先頭となっているため、プレーヤは、そこからデータを読み出してデコードし、素早く画像を表示することができる。

【0097】

なお、このCPIに格納される、デコード単位の先頭位置(この例ではGOPの先頭位置)を、EP (Entry Point)エントリと称する。

【0098】

図19において、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blkCPI()の最後までデータ長を示す。次のif文に従い、フィールドLengthの値が0でなければ、12ビットのデータ長を有する領域reservedを介してフィールドCPITypeが配される。フィールドCPITypeは、4ビットのデータ長を有し、CPIの種類を示す。次のブロックblkEPMa()は、対応するクリップAVストリームファイル

におけるPTS値とバイトアドレスとの関連付けを行うテーブルが格納される。

【0099】

図20は、ブロックblkEMap()の一例の構造を表すシンタクスを示す。8ビットのデータ長を有する領域reservedを介してフィールドNumberOfStreamPIDEntriesが配される。フィールドNumberOfStreamPIDEntriesは、8ビットのデータ長を有し、ブロックblkEMap()におけるブロックblkEMapForOneStreamPIDのエントリ数を示す。forループ文に従い、値[k]をループ変数として、フィールドNumberOfStreamPIDEntriesに示される数だけ、エントリポイントに関する情報が記述される。

【0100】

forループ文内において、フィールドStreamPID[k]は、16ビットのデータ長を有し、ブロックblkEMap()の中で[k]番目にエントリされるブロックblkEMapForOneStreamPID(以下、[k]番目のブロックblkEMapForOneStreamPIDと記述する)によって参照されるエレメンタリストリームを伝送するトランスポートパケットのPIDの値を示す。

【0101】

10ビットのデータ長を有する領域reservedを介してフィールドEPStreamType[k]が配される。フィールドEPStreamType[k]は、4ビットのデータ長を有し、[k]番目のブロックblkEMapForOneStreamPIDによって参照されるエレメンタリストリームのタイプを示す。フィールドNumberOfEPCoarseEntries[k]は、16ビットのデータ長を有し、[k]番目のブロックblkEMapForOneStreamPIDの中にある粗い検索用のサブテーブル(EP coarse table)のエントリ数を示す。フィールドNumberOfEPFineEntries[k]は、18ビットのデータ長を有し、[k]番目のブロックblkEMapForOneStreamPIDの中にある精密な検索用のサブテーブル(EP fine table)のエントリ数を示す。フィールドEMapForOneStreamPIDStartAddress[k]は、32ビットのデータ長を有し、ブロックblkEMap()の中で[k]番目のブロックblkEMapForOneStreamPIDが始まる相対バイト位置を示す。この値は、ブロックblkEMap()の第1バイト目からのバイト数で示される。

【0102】

上述のforループ文による記述の後、16ビットの整数倍のデータ長を有するパディングワードを挟んで記述されるforループ文に従い、値[k]をループ変数として、フィールドNumberOfStreamPIDEntriesに示される数だけ、ブロックblkEMapForOneStreamPID(EPStreamType[k], NumberOfEPCoarseEntries[k], NumberOfEPFineEntries[k])が格納される。すなわち、引数NumberOfEPCoarseEntries[k]は、サブテーブル(EP coarse table)に格納されるエントリPTSEPCoarseおよびエントリSPNEPCoarseの数を示す。同様に、引数NumberOfEPFineEntries[k]は、サブテーブル(EP fine table)に格納されるエントリPTSEPFineおよびエントリSPNEPFineの数を示す。以下では、引数NumberOfEPCoarseEntries[k]および引数NumberOfEPFineEntries[k]を、それぞれ適宜、エントリ数Ncおよびエントリ数Nfと呼ぶ。

【0103】

図21は、ブロックblkEMapForOneStreamPID(EP_stream_type, Nc, Nf)の一例の構造を表すシンタクスを示す。ブロックblkEMapForOneStreamPID(EP_stream_type, Nc, Nf)のセマンティクスを説明するために、まず、ブロックblkEMapForOneStreamPID(EP_stream_type, Nc, Nf)に格納されるデータの元となるエントリである、エントリPTSEPStartおよびエントリSPNEPStartの意味について説明する。

【0104】

エントリPTSEPStartと、エントリPTSEPStartに関連付けられたエントリSPNEPStartは、それぞれAVストリーム上のエントリポイントを指す。そして、エントリPTSEPFineと、エントリPTSEPFineに関連付けられたエントリPTSEPCoarseは、同一のエントリPTSEPStartから導かれる。また、エントリSPNEPFineと、エントリSPNEPFineに関連付けられたエントリSPNEPCoarseは、同一のエントリSPNEPStartから導かれる。

【0105】

図22は、エントリPTSEPCoarseおよびエントリPTSEPFineの一例のフォーマットについ

10

20

30

40

50

て示す。PTSすなわちエントリPTSEPStartは、データ長が33ビットの値である。MSBのビットを第32ビット、LSBのビットを第0ビットとすると、この図22の例では、大まかな単位で検索を行う際に用いられるエントリPTSEPCoarseは、エントリPTSEPStartの第32ビットから第19ビットまでの14ビットが用いられる。エントリPTSEPCoarseにより、解像度が5.8秒で、26.5時間までの範囲で検索が可能である。また、より精密な検索を行うためのエントリPTSEPFineは、エントリPTSEPStartの第19ビットから第9ビットまでの11ビットが用いられる。エントリPTSEPFineにより、解像度が5.7ミリ秒で、11.5秒までの範囲で検索が可能である。なお、第19ビットは、エントリPTSEPCoarseとエントリPTSEPFineとで共通して用いられる。また、LSB側の第0ビットから第8ビットまでの9ビットは、用いられない。

10

【0106】

図23は、エントリSPNEPCoarseおよびエントリSPNEPFineの一例のフォーマットについて示す。ソースパケット番号すなわちエントリSPNEPStartは、データ長が32ビットの値である。MSBのビットを第31ビット、LSBのビットを第0ビットとすると、この図23の例では、大まかな単位で検索を行う際に用いられるエントリSPNEPCoarseは、エントリSPNEPStartの第31ビットから第0ビットまでの全てのビットが用いられる。また、より精密な検索を行うためのエントリSPNEPFineは、エントリSPNEPStartの第16ビットから第0ビットまでの17ビットが用いられる。エントリSPNEPFineにより、例えば略25MB(Mega Byte)のAVストリームファイルまでの範囲で、検索が可能である。

20

【0107】

なお、ソースパケット番号の場合でも、エントリSPNEPCoarseとしてMSB側の所定ビット数の値だけ用いるようにしてもよい。例えば、エントリSPNEPCoarseとして、エントリSPNEPStartの第31ビットから第16ビットまでの17ビットを用い、エントリSPNEPFineは、エントリSPNEPStartの第16ビットから第0ビットまでの17ビットを用いる。

【0108】

上述に基づき、エントリPTSEPStartおよびエントリSPNEPStartは、次のように定義される。

【0109】

エントリPTSEPStartは、図22で示したように、データ長が33ビットの符号無し整数であり、AVストリーム中で、ランダムアクセスが可能なピクチャ(例えばIDR(Instantaneous Decoding Refresh)ピクチャやI(Intra)ピクチャ)から開始するビデオアクセスユニットの33ビット長のPTSを示す。

30

【0110】

エントリSPNEPStartは、図23で示したように、32ビットの符号無し整数であり、エントリPTSEPStartに関連付けられたビデオアクセスユニットの第1バイト目を含むソースパケットの、AVストリームの中でのアドレスを示す。エントリSPNEPStartは、ソースパケット番号の単位で表され、AVストリームファイル中の最初のソースパケットから、値"0"を初期値として、ソースパケット毎に1ずつ増加する値としてカウントされる。

【0111】

図21を参照し、ブロックblkEMapForOneStreamPID(EP_stream_type, Nc, Nf)は、第1のforループ文により大まかな単位での検索を行うためのサブテーブル(EP coarse table)が記述され、第2のforループ文によりサブテーブル(EP coarse table)の検索結果に基づきより詳細な検索を行うためのサブテーブル(EP fine table)が記述される。

40

【0112】

第1のforループ文の直前に、フィールドEPFineTableStartAddressが配される。フィールドEPFineTableStartAddressは、32ビットのデータ長を有し、最初の第2のforループにおけるフィールドReservedEPFine[EP_fine_id]の第1バイト目の開始アドレスを、ブロックblkEMapForOneStreamPID(EP_stream_type, Nc, Nf)の第1バイト目からの相対バイト数で示す。相対バイト数は、値"0"から開始する。

【0113】

50

第1のforループ文は、ループ変数[i]で以て、サブテーブル(EP coarse table)のエントリ数Ncまで繰り返され、エントリ数Ncの組数だけフィールドRefToEPFineID[i]、エントリPTSEPCoarse[i]およびエントリPTSEPFine[i]が格納される。第1のforループ文において、フィールドRefToEPFineID[i]は、18ビットのデータ長を有し、フィールドRefToEPFineID[i]に続くフィールドPTSEPCoarse[i]が示すエントリPTSEPCoarseに関連付けられるエントリPTSEPFineを持つ、サブテーブル(EP fine table)内のエントリ番号を示す。エントリPTSEPFineと、このエントリPTSEPFineに関連付けられるエントリPTSEPCoarseとは、同一のエントリPTSEPStartから導かれる。フィールドRefToEPFineID[i]は、第2のforループ文中で記述される順番で定義されるループ変数[EP_fine_id]の値により与えられる。

【0114】

10

第1のforループ文の後に、パディングワードを挟んで第2のforループ文による記述がなされる。第2のforループ文は、ループ変数[EP_fine_id]で以て、サブテーブル(EP fine table)のエントリ数Nfまで繰り返され、エントリ数Nfの組数だけ、1ビットのデータ長を有するフィールドReservedEPFine[EP_fine_id]と、3ビットのデータ長を有するフィールドLEndPositionOffset[EP_fine_id]と、11ビットのデータ長を有するフィールドPTSEPFine[EP_fine_id]と、17ビットのデータ長を有するフィールドSPNEPFine[EP_fine_id]とが格納される。これらのうち、フィールドPTSEPFine[EP_fine_id]およびフィールドSPNEPFine[EP_fine_id]は、ループ変数[EP_fine_id]に基づきサブテーブル(EP fine table)から参照されるエントリPTSEPFineおよびエントリSPNEPFineそれぞれが格納される。

20

【0115】

エントリPTSEPCoarseおよびエントリPTSEPFine、ならびに、エントリSPNEPCoarseおよびエントリSPNEPFineは、次のように導かれる。サブテーブル(EP fine table)に、関連するデータSPNEPStartの値の昇順に並んでいるNf個のエントリがあるとする。それぞれのエントリPTSEPFineは、対応するエントリPTSEPStartから、次式(1)のように導かれる。

$$PTSEPFine[EP_fine_id] = (PTSEPStart[EP_fine_id] \gg 9) / 2^{11} \cdots (1)$$

【0116】

エントリPTSEPCoarseと、対応するエントリPTSEPFineとの関係は、次式(2)、(3)の通りである。

$$PTSEPCoarse[i] = (PTSEPStart[RefToEPFineID[i]] \gg 19) / 2^{14} \cdots (2)$$

$$PTSEPFine[RefToEPFineID[i]] = (PTSEPStart[RefToEPFineID[i]] \gg 9) / 2^{11} \cdots (3)$$

30

【0117】

それぞれのエントリSPNEPFineは、対応するエントリSPNEPStartから、次式(4)のように導かれる。

$$SPNEPFine[EP_fine_id] = SPNEPStart[EP_fine_id] / 2^{17} \cdots (4)$$

【0118】

エントリSPNEPCoarseと、対応するエントリSPNEPFineとの関係は、次式(5)、(6)の通りである。

$$SPNEPCoarse[i] = SPNEPStart[RefToEPFineID[i]] \cdots (5)$$

$$SPNEPFine[RefToEPFineID[i]] = SPNEPStart[RefToEPFineID[i]] / 2^{17} \cdots (6)$$

40

【0119】

なお、上述の式(1)~(6)において、記号「 $\gg x$ 」は、データのLSB側からxビットを超える桁からビットを用いることを意味する。

【0120】

次に、拡張データを格納するためのブロックblkExtensionData()について説明する。このブロックblkExtensionData()は、所定の拡張データを格納可能なように定義され、インデックステーブルが格納されるファイル"index.bdmv"、プレイリストが格納されるファイル"xxxxx.mpls"およびクリップインフォメーションファイル"zzzzz.clpi"の各ファイルに記述することができる。

【0121】

50

図 2 4 は、ブロックblkExtensionData()の一例の構造を表すシンタクスを示す。フィールドLengthは、3 2 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。このフィールドLengthの示すデータ長が" 0 "でなければ、if文以下の記述がなされる。

【 0 1 2 2 】

フィールドDataBlockStartAddressは、3 2 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。すなわち、相対バイト数は、" 0 "から開始される。なお、フィールドDataBlockStartAddressは、次に示す 3 2 ビットアライメントの条件を満たさなければならない。

DataBlockStartAddress % 4 = 0

【 0 1 2 3 】

2 4 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。この例では、拡張データのエントリは、フィールドExtDataType、フィールドExtDataVersion、フィールドExtDataStartAddressおよびフィールドExtDataLengthからなるブロックext_data_entry()であって、ブロックblkExtensionData()において、第 1 のforループ文に従いこのフィールドNumberOfExtDataEntriesに示される個数だけ、このブロックext_data_entry()が存在する。

【 0 1 2 4 】

フィールドExtDataTypeは、1 6 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()を含む規格書のライセンサ（使用認可者）が割り当てると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。なお、このブロックblkExtensionData()において、フィールドExtDataTypeおよびフィールドExtDataVersionの値が同一のブロックext_data_entry()が 2 以上、存在してはならない。

【 0 1 2 5 】

フィールドExtDataStartAddressは、3 2 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。なお、フィールドExtDataStartAddressは、次に示す 3 2 ビットアライメントの条件を満たさなければならない。

ExtDataStartAddress % 4 = 0

【 0 1 2 6 】

フィールドExtDataLengthは、3 2 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。データ長は、バイト数で示される。

【 0 1 2 7 】

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 1 6 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。

【 0 1 2 8 】

図 2 5 は、ブロックblkExtensionData()における各データの参照関係を模式的に示す。フィールドLengthにより、フィールドLength直後の位置からブロックblkExtensionData()の最後までデータ長が示される。フィールドDataBlockStartAddressにより、ブロックDataBlock()の開始位置が示される。フィールドNumberOfExtDataEntriesで示される個数だけ、ブロックext_data_entryが記述される。最後のブロックext_data_entryからブロックDataBlock()の間には、任意の長さでフィールドpadding_wordが置かれる。

【 0 1 2 9 】

ブロックDataBlock()内には、ブロックext_data_entry()で示される拡張データext_dataが置かれる。それぞれの拡張データext_dataの位置およびデータ長は、対応するブロックext_data_entry()内のフィールドExtDataStartAddressおよびフィールドExtDataLengthにより示される。したがって、ブロックDataBlock()内での拡張データext_dataの並び順は、対応するブロックext_data_entry()の並び順と一致していなくてもよい。

【 0 1 3 0 】

このように、拡張データを、拡張データの本体が格納されるブロックDataBlock()と、ブロックDataBlock()内の拡張データに対するアクセス情報などが格納されるブロックext_data_entry()とによる 2 層構造とすることで、複数の拡張データを格納することが可能となる。

【 0 1 3 1 】

次に、上述の拡張データの一例の作成方法および読み出し方法について説明する。図 2 6 は、ブロックblkExtensionData()にデータを書き込む際の一例の処理を示すフローチャートである。この図 2 6 は、ブロックblkExtensionData()中の (n + 1) 番目のエントリとして、拡張データを追加し、ブロックblkExtensionData()を書き換える場合の例である。

【 0 1 3 2 】

まず、ステップ S 1 0 で、書き込もうとしている拡張データのデータ長を取得し、フィールドExtDataLength[n+1]の値にセットする。なお、「[n+1]」の記述は、(n + 1) 番目のエントリの番号に対応する。次に、ステップ S 1 1 で、現在のブロックblkExtensionData()に列挙されているブロックext_data_entry()のフィールドExtDataLengthおよびフィールドExtDataStartAddressの値を調べ、ブロックDataBlock()の使用状況を取得する。

【 0 1 3 3 】

そして、次のステップ S 1 2 で、ブロックDataBlock()中に、書き込もうとしている拡張データのデータ長であるフィールドExtDataLength[n+1]に示されるデータ長以上の、連続した空き領域があるか否かが判断される。若し、あると判断されれば、処理はステップ S 1 4 に移行される。

【 0 1 3 4 】

一方、フィールドExtDataLength[n+1]に示されるデータ長以上の連続した空き領域が無いと判断されれば、処理はステップ S 1 3 に移行され、ブロックblkExtensionData()におけるフィールドLengthの値を大きくし、フィールドExtDataLength[n+1]に示されるデータ長以上の連続した空き領域をブロックDataBlock()内に作る。空き領域ができたなら、処理がステップ S 1 4 に移行される。

【 0 1 3 5 】

ステップ S 1 4 では、拡張データを書き込む領域の先頭アドレスを決め、その先頭アドレスの値をフィールドExtDataStartAddress[n+1]とする。次のステップ S 1 5 で、フィールドExtDataStartAddress[n+1]から、上述のステップ S 1 0 でセットされたフィールドExtDataLength[n+1]の長さの拡張データext_data[n+1]を書き込む。

【 0 1 3 6 】

データの書き込みが終了したら、ステップ S 1 6 で、ブロックext_data_entry()に対して、フィールドExtDataLength[n+1]と、フィールドExtDataStartAddress[n+1]とを追加する。

10

20

30

40

50

【 0 1 3 7 】

なお、上述において、書き換えを行うブロックblkExtensionData()は、すでにディスクなどの記録媒体から読み出されて記録装置のメモリに記憶されているものとする。そのため、ステップS 1 3における、フィールドLengthの値の変更によるブロックblkExtensionData()の拡大は、システムに任せ、システムがメモリアロケーションを適切に行うことでなされる。

【 0 1 3 8 】

図 2 7 は、ブロックblkExtensionData()から拡張データを読み出す際の一例の処理を示すフローチャートである。なお、この図 2 7 のフローチャートによる処理は、再生専用の記録媒体と、記録可能な記録媒体との両方に適用可能なものである。まず、最初のステップS 2 0 で、読み込もうとする拡張データが準拠する規格から、フィールドExtDataTypeの値を取得し、ステップS 2 1 で、読み込もうとする拡張データの種別から、フィールドExtDataVersionの値を取得する。

10

【 0 1 3 9 】

次のステップS 2 2 で、ブロックblkExtensionData()に列挙されているブロックext_data_entry()を 1 つずつ順次、読み込む。そして、ステップS 2 3 で、読み込んだブロックext_data_entry()に含まれるフィールドExtDataTypeおよびフィールドExtDataVersionの値が、上述のステップS 2 0 およびステップS 2 1 で取得したフィールドExtDataTypeおよびフィールドExtDataVersionの値と一致するか否かが判断される。

20

【 0 1 4 0 】

一致していないと判断されれば、処理はステップS 2 6 に移行され、ブロックblkExtensionData()内に列挙されるブロックext_data_entry()を全て読み終えたか否かが判断される。全て読み終えたと判断されれば、処理はステップS 2 7 に移行され、このブロックblkExtensionData()には、読み込もうとした拡張データが存在しないとして、一連の処理が終了される。全て読み終わっていないと判断されれば、処理はステップS 2 2 に戻され、次のブロックext_data_entry()が読み込まれる。

【 0 1 4 1 】

上述のステップS 2 3 において、ブロックext_data_entry()に含まれるフィールドExtDataTypeおよびフィールドExtDataVersionの値が、取得したフィールドExtDataTypeおよびフィールドExtDataVersionの値と一致していると判断されれば、処理はステップS 2 4 に移行される。ここでは、ブロックblkExtensionData()中の[i]番目のエントリで一致したものとする。

30

【 0 1 4 2 】

ステップS 2 4 では、[i]番目のエントリのブロックext_data_entry()からフィールドExtDataLength[i]の値と、フィールドExtDataStartAddress[i]の値とを読み込む。そして、ステップS 2 5 で、ステップS 2 4 で読み込んだフィールドExtDataStartAddress[i]で示されるアドレスから、フィールドExtDataLength[i]で示されるデータ長だけ、データを読み出す。

【 0 1 4 3 】

次に、上述した、インデックスファイル"index.bdmv"、ムービーオブジェクトファイル"MovieObject.bdmv"、プレイリストファイル"xxxxx.mpls"およびクリップインフォメーションファイル"zzzzz.clpi"にそれぞれ定義可能な、拡張データを格納する拡張データブロックblkExtensionData()について説明する。

40

【 0 1 4 4 】

まず、インデックスファイル"index.bdmv"に対して定義される一例の拡張データブロックについて説明する。ここでは、プレイリスト毎に記録可能な記録媒体に特有の属性情報を付加するようにした、一例の拡張データブロックについて説明する。図 2 8 は、このプレイリスト属性を記述するための、ファイル"index.bdmv"内のフィールドblkExtensionData()におけるブロックDataBlock() (図 2 4 参照) の一例の構造を表すシンタックスを示す。この図 2 8 の例では、ブロックDataBlock()がブロックblkIndexExtensionData()として

50

記述されている。

【 0 1 4 5 】

先ず、上述の図 2 4 を参照して、ブロックblkExtensionData()においてフィールドExtDataTypeを値"0x1000"、フィールドExtDataVersionを値"0x0100"とする。これらフィールドExtDataTypeおよびフィールドExtDataVersionに記述された値は、例えば再生装置側において、予めROM(Read Only Memory)などに記憶されたテーブルが参照されて識別される。ブロックDataBlock()内のフィールドExtDataStartAddressおよびフィールドExtDataLengthで示される領域に、ブロックblkIndexExtensionData()が格納される。なお、数値の記述において"0x"は、その数値が16進表記されていることを示す。

【 0 1 4 6 】

ブロックblkIndexExtensionData()において、フィールドTypeIndicatorは、次に続くデータの種類を示す、ISO 646に規定された符号化方式で符号化した4文字からなる文字列が記述される。この図 2 8 の例では、フィールドTypeIndicatorにISO 646に既定の方式で符号化された4文字の文字列"IDEX"が記述され、次に続くデータ種類が「IndexExtensionData」であることが示される。

【 0 1 4 7 】

フィールドTypeIndicatorに続けて32ビットのデータ長を有する領域reserved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TableOfPlayListStartAddressが配される。フィールドTableOfPlayListStartAddressは、ブロックblkTableOfPlayList()の、このブロックblkIndexExtensionData()先頭を基準とした開始アドレスが示される。

【 0 1 4 8 】

フィールドTableOfPlayListStartAddressの次に、32ビットのデータ長を有するフィールドMakersPrivateDataStartAddressが配されブロックblkMakersPrivateData()のこのブロックblkIndexExtensionData()先頭を基準とした開始アドレスが示され、192ビットのデータ長を有する領域reservedを介してブロックblkUIAppInfoAVCHD()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blkTableOfPlayLists()が配される。さらに続けて、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。このブロックblkMakersPrivateData()の後に、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N3で示される回数だけ繰り返される。

【 0 1 4 9 】

なお、ブロックblkUIAppInfoAVCHD()およびブロックblkMakersPrivateData()は、この発明と関連性が薄いので、説明を省略する。

【 0 1 5 0 】

図 2 9 は、上述したブロックblkTableOfPlayLists()の一例の構造を表すシンタクスを示す。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blkTableOfPlayList()の最後のバイトまでのデータ長をバイト数で示す。フィールドLengthに続けて、プレイバックタイトルを再生するためのプレイリストに関する情報が記述されるブロックblkFirstPlaybackTitlePlayLists()と、メニュータイトルに関する情報が記述されるブロックblkMenuTitlePlayLists()とが配される。これらブロックblkFirstPlaybackTitlePlayLists()およびブロックblkMenuTitlePlayLists()は、この発明と関連性が薄いので、説明を省略する。

【 0 1 5 1 】

次に、16ビットのデータ長を有するフィールドNumberOfTitlePlayListPairが配される。フィールドNumberOfTitlePlayListPairは、プレイバックタイトルおよびメニュータイトル以外のタイトルを再生するためのプレイリストの数が記述される。次のforループ文に従い、フィールドNumberOfTitlePlayListPairで示される数だけ、ブロックblkMovieTitlePlayListPair()が記述される。ブロックblkMovieTitlePlayListPair()は、フィールドPlayListFileName、フィールドPlayListAttributeおよびフィールドRefToTitleIDを含

10

20

30

40

50

む。すなわち、ブロックblkMovieTitlePlayListPair()は、このforループ文で示される[i]番目のプレイリストについて、当該プレイリストのファイル名、当該プレイリストに付与された属性、ならびに、当該プレイリストの参照タイトルIDからなるプレイリストの情報を構造化したものである。

【0152】

このforループ文による並び順は、記録順とされる。すなわち、1のプレイリストが追加されると、フィールドNumberOfTitlePlayListPairの値が1だけインクリメントされ、既存のプレイリストの情報の後ろに、追加されたプレイリストの情報が追記される。

【0153】

フィールドPlayListFileNameは、40ビット(5バイト)のデータ長を有し、プレイリストのファイル名がISO646に規定された符号化方式で符号化されて記述される。フィールドPlayListFileNameの次に、6ビットのデータ長を有する領域reservedを介してフィールドPlayListAttributeが配される。フィールドPlayListAttributeは、2ビットのデータ長を有し、当該プレイリストに付与された属性を示す。プレイリストは、その成因に基づき、クリップの生成と共に生成されるプレイリストに対応する第1の種類と、既存のタイトルあるいはプレイリストの一部または全部を用いて作成されるプレイリストに対応する第2の種類と、メニューを再生するために用いる第3の種類との3種類に分けられ、各プレイリストには、プレイリストの種類に応じて、それぞれ対応する属性「Real」(第1の種類)、属性「Virtual」(第2の種類)および属性「Menu」(第3の種類)が付与される。

【0154】

なお、以下では適宜、属性「Real」が付与されたプレイリストをリアルプレイリスト、属性「Virtual」が付与されたプレイリストをバーチャルプレイリスト、属性「Menu」を付与されたプレイリストをメニュープレイリストと呼ぶ。

【0155】

フィールドRefToTitleIdは、同ループ内のフィールドPlayListFileNameに示されるプレイリストが作成時に属するタイトルのID(番号)が記述される。より具体的な例としては、インデックスファイル"index.bdmv"内のブロックblkIndexes()における、対応する値title_idが記述される。なお、当該プレイリストがファーストプレイバックタイトルのみから再生される場合、フィールドRefToTitleIdの値は、第1の固定値、例えば"0xFFFF"とされる。また、当該プレイリストがメニュータイトルのみから再生される場合は、フィールドRefToTitleIdの値は、第2の固定値、例えば"0xFFFE"とされる。

【0156】

次に、仮想プレーヤについて、概略的に説明する。上述したようなデータ構造を有するディスクがプレーヤに装填されると、プレーヤは、ディスクから読み出されたムービーオブジェクトなどに記述されたコマンドを、プレーヤ内部のハードウェアを制御するための固有のコマンドに変換する必要がある。プレーヤは、このような変換を行うためのソフトウェアを、プレーヤに内蔵されるROM(Read Only Memory)に予め記憶している。このソフトウェアは、ディスクとプレーヤを仲介してプレーヤにAVCHDフォーマットの規定に従った動作をさせることから、仮想プレーヤと称される。

【0157】

図30は、この仮想プレーヤの動作を概略的に示す。図30Aは、ディスクのローディング時の動作の例を示す。ディスクがプレーヤに装填されディスクに対するイニシャルアクセスがなされると(ステップS30)、1のディスクにおいて共有的に用いられる共有パラメータが記憶されるレジスタが初期化される(ステップS31)。そして、次のステップS32で、ムービーオブジェクトなどに記述されたプログラムがディスクから読み込まれて実行される。なお、イニシャルアクセスは、ディスク装填時のように、ディスクの再生が初めて行われることをいう。

【0158】

図30Bは、プレーヤが停止状態からユーザにより例えばプレイキーが押下され再生が

10

20

30

40

50

指示された場合の動作の例を示す。最初の停止状態（ステップS40）に対して、ユーザにより、例えばリモートコントロールコマンドなどを用いて再生が指示される（UO: User Operation）。再生が指示されると、先ず、レジスタすなわち共通パラメータが初期化され（ステップS41）、次のステップS42で、ムービーオブジェクト実行フェイズに移行する。

【0159】

ムービーオブジェクトの実行フェイズにおけるプレイリストの再生について、図31を用いて説明する。UOなどにより、タイトル番号#1のコンテンツを再生開始する指示があった場合について考える。プレーヤは、コンテンツの再生開始指示に応じて、上述した図2に示されるインデックステーブル(Index Table)を参照し、タイトル#1のコンテンツ再生に対応するオブジェクトの番号を取得する。例えばタイトル#1のコンテンツ再生を実現するオブジェクトの番号が#1であったとすると、プレーヤは、ムービーオブジェクト#1の実行を開始する。

10

【0160】

この図31の例では、ムービーオブジェクト#1に記述されたプログラムは2行からなり、1行目のコマンドが"Play PlayList(1)"であるとする、プレーヤは、プレイリスト#1の再生を開始する。プレイリスト#1は、1以上のプレイアイテムから構成され、プレイアイテムが順次再生される。プレイリスト#1中のプレイアイテムの再生が終了すると、ムービーオブジェクト#1の実行に戻り、2行目のコマンドが実行される。図31の例では、2行目のコマンドが"jump MenuTitle"であって、このコマンドが実行されインデックステーブルに記述されたメニュータイトル(MenuTitle)を実現するムービーオブジェクトの実行が開始される。

20

【0161】

次に、この発明の実施の一形態について説明する。この発明では、映像信号の記録開始から記録終了までを1つの記録単位として定義し、この1記録単位の区間に生成されるビデオデータからなるストリームを1つのストリームファイルとして記録媒体に記録する。このストリームファイルには、映像信号の記録に伴い記録される音声信号を含むことができる。この1記録単位に対応するストリームファイルに基づきクリップが構成される。

【0162】

また、この発明の実施の一形態では、1つのクリップを1つのプレイアイテムから参照し、クリップが生成される毎に、生成されたクリップを参照するプレイアイテムを順次、プレイリストに対して追加する。このとき、プレイリストにおいて、プレイアイテム毎すなわちクリップ毎にプレイリストマークを打つ。そして、例えば1の記録媒体に1のプレイリストのみが存在するようにする。これに限らず、ディスク上に複数のプレイリストを存在させてもよい。

30

【0163】

図32は、この発明の実施の一形態に適用可能な記録装置の一例の構成を概略的に示す。この記録装置は、入力されたデジタルビデオデータおよびデジタルオーディオデータを、所定的方式で圧縮符号化および多重化したAVストリームを記録媒体に記録するようにしている。圧縮符号化および多重化の方式としては、例えば上述したAVCHDフォーマットが適用できる。

40

【0164】

この図32に例示される記録装置は、外部から入力されるビデオデータおよびオーディオデータを記録媒体に記録する、単独の記録装置として用いることもできるし、光学系や撮像素子などを備えたカメラブロックと組み合わせ、撮像した撮像信号に基づくビデオデータを記録媒体に記録する、ビデオカメラ装置の記録ブロックとして用いることもできる。

【0165】

適用可能な圧縮符号化や多重化の方式としては、様々に考えられる。例えば、H.264 | AVCに規定される方式を、この発明の実施の一形態の圧縮符号化として適用するこ

50

とができる。これに限らず、MPEG2方式に基づき圧縮符号化を行うようにしてもよい。また、多重化方式は、例えばMPEG2システムズを適用することができる。以下では、ビデオデータの圧縮符号化をH.264|AVCに規定される方式に準じて行い、ビデオデータおよびオーディオデータの多重化を、MPEG2システムズに規定される方式に準じて行うものとして説明する。

【0166】

制御部30は、例えばCPU(Central Processing Unit)、RAM(Random Access Memory)およびROM(Read Only Memory)などからなり(図示しない)、ROMに予め記憶されたプログラムやデータに基づき、RAMをワークメモリとして用いてこの記録装置の記録部10の各部を制御する。なお、制御部10と記録部10の各部とを接続する経路は、
10

【0167】

UI(User Interface)部31は、この記録装置の動作をユーザが操作するための操作子が所定に設けられ、操作子に対する操作に応じた制御信号を出力する。この制御信号は、制御部30に供給される。制御部30は、ユーザ操作に応じてUI部31から供給された制御信号に基づきなされるプログラムの処理により、記録部10の各部の動作を制御する。例えば、UI部31に対してなされた操作に応じて、記録装置による記録動作の開始および停止の動作が制御部30により制御される。

【0168】

ベースバンドのデジタルビデオデータが端子40から入力される。また、当該デジタルビデオデータに伴い、ベースバンドのデジタルオーディオデータが端子41から入力される。
20

【0169】

デジタルビデオデータは端子40から記録部10に入力され、ビデオエンコーダ11に供給される。ビデオエンコーダ11は、供給されたデジタルビデオデータを、所定の方式で以て圧縮符号化する。MPEG4|AVC|H.264に規定される方式に準じて圧縮符号化がなされるこの例では、例えば、DCT(Discrete Cosine Transform)と画面内予測とによりフレーム内圧縮を行うと共に、動きベクトルを用いたフレーム間圧縮を行い、さらにエントロピー符号化を行い圧縮効率を高める。ビデオエンコーダ11で圧縮符号化されたデジタルビデオデータは、MPEG4のエレメンタリストリーム(ES)として、マルチプレクサ(MUX)13に供給される。
30

【0170】

デジタルオーディオデータは端子41から記録部10に入力され、オーディオエンコーダ12に供給される。オーディオエンコーダ12は、所定の圧縮符号化方式、例えばAAC(Advanced Audio Coding)により圧縮符号化される。オーディオデータの圧縮符号化方式は、AAC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。圧縮符号化されたデジタルオーディオデータは、マルチプレクサ13に供給される。

【0171】

マルチプレクサ13は、それぞれ圧縮符号化されて供給されたデジタルビデオデータおよびデジタルオーディオデータを所定の方式で多重化し、1本のデータストリームとして出力する。MPEG2システムズに準じて多重化が行われるこの例では、MPEG2のトランスポートストリームを用いて、供給された圧縮ビデオデータおよび圧縮オーディオデータを時分割で多重化する。例えば、マルチプレクサ13は、バッファメモリを有し、供給された圧縮ビデオデータおよび圧縮オーディオデータをバッファメモリに溜め込む。
40

【0172】

バッファメモリに溜め込まれた圧縮ビデオデータは、所定サイズ毎に分割されヘッダが付加されて、PES(Packetized Elementary Stream)パケット化される。圧縮オーディオデータも同様に、所定サイズ毎に分割されヘッダが付加されてPESパケット化される。
50

ヘッダには、パケットに格納されるデータの再生時刻を示すPTSや復号時刻を示すDTS(Decoding Time Stamp)といった、MPEG2システムズに規定される所定の情報が格納される。PESパケットは、さらに分割されてトランスポートパケット(TSパケット)のペイロードに詰め込まれる。TSパケットのヘッダには、ペイロードに詰め込まれたデータを識別するためのPID(Packet Identification)が格納される。マルチプレクサ13から出力されたTSパケットは、ストリームバッファ14に一旦溜め込まれる。

【0173】

なお、TSパケットは、実際には、マルチプレクサ13においてさらに所定サイズのヘッダが付加されて出力される。この、TSパケットに対して所定のヘッダを付加したパケットを、ソースパケットと呼ぶ。

【0174】

記録制御部15は、記録媒体20に対するデータの記録を制御する。記録媒体20としては、例えば記録可能なタイプのDVD(Digital Versatile Disc)を用いることができる。これに限らず、記録媒体20としてハードディスクドライブを用いてもよいし、半導体メモリを記録媒体20に適用することも可能である。また、記録媒体20として、より大容量を実現したBlu-ray Disc(ブルーレイディスク：登録商標)を適用することも考えられる。

【0175】

記録制御部15は、ストリームバッファ14に溜め込まれたデータ量を監視し、ストリームバッファ14に所定量以上のデータが溜め込まれると、ストリームバッファ14から記録媒体20の記録単位分のデータを読み出して記録媒体20に書き込む。

【0176】

管理情報処理部16は、例えばCPU、ワークメモリとしてのRAMおよびプログラム所定のデータが予め記憶されるROMからなる(図示しない)。これに限らず、管理情報処理部16は、例えば制御部30におけるプログラム処理で管理情報処理部16の機能を実現することも可能である。この場合、例えば制御部30の有するRAMが揮発性メモリ17として用いられると共に、不揮発性メモリ18が制御部30に接続される。

【0177】

管理情報処理部16は、記録データに基づき、揮発性メモリ17をワークメモリとして用いて、上述したインデックスファイル"index.bdmv"、ムービーオブジェクトファイル"MovieObject.bdmv"、プレイリストファイル"xxxxx.mpls"およびクリップインフォメーションファイル"zzzzz.clpi"に格納するための情報を生成する。生成された情報は、所定のタイミングで記録媒体20に書き込まれる。

【0178】

一例として、管理情報処理部16は、マルチプレクサ13から記録データの時間情報を取得すると共に、記録制御部15から記録データの記録媒体20に対するアドレス情報を取得し、取得されたこれら時間情報およびアドレス情報に基づきEP_map情報が生成される。また、UI部31に対する記録開始、記録終了の操作に応じて制御部30から出力される制御信号と、マルチプレクサ13および記録制御部15からの記録データに関する情報とに基づき、プレイリストファイル"xxxxx.mpls"の生成または更新、クリップインフォメーションファイル"zzzzz.clpi"の生成などが行われる。さらに、記録媒体20に対して新規に記録が行われる際には、インデックスファイル"index.bdmv"やムービーオブジェクトファイル"MovieObject.bdmv"の生成または更新が行われる。

【0179】

次に、この発明の実施の一形態によるクリップの記録方法について説明する。図33は、この発明の実施の一形態によるクリップの一例の記録方法を示すフローチャートである。このフローチャートによる処理に先立って、記録媒体20には、クリップと、当該クリップを再生するための再生制御情報が格納されファイル(クリップインフォメーションファイル、プレイリストファイル、ムービーオブジェクトファイルおよびインデックスファイルなど)とが既に記録媒体20上に記録されているものとする。

【 0 1 8 0 】

ステップ S 5 0 で記録開始操作が行われると、次のステップ S 5 1 で、クリップ A V ストリームの記録媒体 2 0 への記録が開始される。

【 0 1 8 1 】

記録開始操作は、例えば次のようになされる。例えば、U I 部 3 1 に、記録開始を指示する記録開始スイッチと、記録停止を指示する記録停止スイッチとが設けられる。ステップ S 5 0 では、ユーザによって記録開始スイッチが操作される。この操作に応じて、記録開始を指示する制御信号が U I 部 3 1 から出力され、制御部 3 0 に供給される。制御部 3 0 は、この記録開始を指示する制御信号に基づき、記録部 1 0 の各部に対し、端子 4 0 から入力されるベースバンドのビデオデータと、端子 4 1 から入力されるベースバンドのオーディオデータとを記録媒体 2 0 に記録するように制御する。

10

【 0 1 8 2 】

記録開始に関わる制御部 3 0 の制御の一例として、記録停止状態ではビデオエンコーダ 1 1 およびオーディオエンコーダ 1 2 の動作を停止させておき、記録開始の指示に応じて、これらビデオエンコーダ 1 1 およびオーディオエンコーダ 1 2 の動作を開始させることが考えられる。これに限らず、マルチプレクサ 1 3 やストリームバッファ 1 4、記録制御部 1 5 の動作の開始、停止を制御することで、記録開始および停止の制御を行うこともできる。

【 0 1 8 3 】

記録開始の制御に応じて、クリップ A V ストリームが記録媒体 2 0 に記録される（ステップ S 5 1）。すなわち、入力されたビデオデータおよびオーディオデータがビデオエンコーダ 1 1 およびオーディオエンコーダ 1 2 でそれぞれ圧縮符号化され、マルチプレクサ 1 3 でパケット化され T S パケット（実際にはさらに所定のヘッダが付加されたソースパケット）とされてストリームバッファ 1 4 に供給される。ストリームバッファ 1 4 に所定量以上の T S パケットが溜め込まれたら、記録制御部 1 5 によりストリームバッファ 1 4 から T S パケットが読み出される。読み出された T S パケットは、所定にファイル名が付されたクリップ A V ストリームファイルに格納されて記録媒体 2 0 に記録される。

20

【 0 1 8 4 】

例えば、記録媒体 2 0 に既にファイル名 "00001.m2ts" であるクリップ A V ストリームファイルが記録されている場合には、新たに記録されるクリップ A V ストリームファイルのファイル名として既に記録されているファイルと重複しないファイル名が選ばれ、例えばファイル名 "00002.m2ts" とされる。

30

【 0 1 8 5 】

なお、クリップ A V ストリームの記録媒体 2 0 への記録に伴い、管理情報処理部 1 6 により、記録されるデータの再生時間とアドレスとの対応関係を示す情報がリアルタイムに生成される。このデータは、上述したクリップインフォメーションファイル "zzzzz.clpi" 内のブロック blkEPMMap() に格納されるデータとして、揮発性メモリ 1 7 に記憶される。当該データのバックアップとして、不揮発性メモリ 1 8 にも同一のデータを記憶することもできる。

【 0 1 8 6 】

次のステップ S 5 2 で、記録停止操作が行われたか否かが判断される。例えば、ユーザにより U I 部 3 1 に設けられた記録停止スイッチが操作され、記録が停止されたと判断されれば、処理はステップ S 5 3 に移行される。一方、記録が停止されていないければ、処理はステップ S 5 1 に戻され、クリップ A V ストリームの記録媒体 2 0 への記録が継続される。

40

【 0 1 8 7 】

ステップ S 5 3 では、記録の停止に伴い、ストリームバッファ 1 4 に溜め込まれているストリームが全て記録媒体 2 0 に書き込まれる。例えば、記録制御部 1 5 は、制御部 3 0 からの記録停止の命令に応じて、ストリームバッファ 1 4 に溜め込まれているストリーム（T S パケット）を全て読み出し、記録媒体 2 0 に書き込む。

50

【 0 1 8 8 】

また、記録停止の命令に応じて、例えばビデオエンコーダ 1 1 およびオーディオエンコーダ 1 2 の動作が停止される。このとき、図 1 3 A を用いて説明した第 1 のシームレス接続を行うために、例えば、オーディオエンコーダ 1 2 の動作がビデオエンコーダ 1 1 の動作が停止してから所定時間後に停止されるように制御される。

【 0 1 8 9 】

次のステップ S 5 4 ~ ステップ S 5 8 で、管理情報処理部 1 6 により、記録媒体 2 0 に書き込まれたクリップ A V ストリームファイルに関するクリップインフォメーションファイルが生成されると共に、プレイリストファイルの更新が成される。

【 0 1 9 0 】

先ず、ステップ S 5 4 で、管理情報処理部 1 6 により、クリップインフォメーションファイル "zzzzz.clpi" が生成される。ファイル名は、例えばこのクリップインフォメーションファイルが示すクリップ A V ストリームファイルのファイル名と対応するファイル名とされ、当該クリップ A V ストリームファイルのファイル名が "00002.m2ts" であれば、このクリップインフォメーションファイルのファイル名は、拡張子より前の部分が同一のファイル名 "00002.clpi" とされる。

【 0 1 9 1 】

クリップインフォメーションファイル "00002.clpi" に、図 1 5 ~ 図 2 1 に例示した各シンタクスに従い、各フィールドやフラグの値が所定に設定され格納される。一例として、TS パケットに関する情報や、再生時間 (PTS) に関する情報は、管理情報処理部 1 6 により、クリップの記録中にマルチプレクサ 1 3 から取得された情報に基づき生成される。また、記録媒体 2 0 上の記録アドレスに関する情報は、管理情報処理部 1 6 により、クリップの記録中に記録制御部 1 5 から取得された情報に基づき生成される。システムにより固有の値は、例えば予め ROM (図示しない) などに記憶されている情報に基づく。さらに、再生時間とアドレスとの対応関係を示す上述したブロック blkEPMAP() の情報が、クリップインフォメーションファイル "00002.clpi" のブロック blkCPI() に格納される。

【 0 1 9 2 】

また、ブロック blkClipInfo() 内のフラグ IsCC5 は、ユーザ操作によりクリップの記録が停止された場合、値が 1 (バイナリ値) とされる。それに伴い、ブロック blkClipInfo() 内の if 文 (図 1 6 参照) で示されるデータが所定に設定される。

【 0 1 9 3 】

クリップインフォメーションファイルの作成が完了したら、処理は次のステップ S 5 5 に移行する。ステップ S 5 5 ~ ステップ S 5 8 の処理は、プレイリストファイルに関する処理である。このステップ S 5 5 ~ ステップ S 5 8 の処理により、既に記録媒体 2 0 上に存在するプレイリストファイルに対して、新たに記録されたクリップ A V ストリームファイル "00002.m2ts" に対応するプレイアイテムが追加される。

【 0 1 9 4 】

先ず、ステップ S 5 5 で、プレイリストファイル内のブロック blkPlayItem() におけるフィールド ConnectionCondition の値が 5 に設定され、このクリップが次のクリップと第 1 のシームレス接続を行うことが示される (図 1 2 参照)。次にステップ S 5 6 で、プレイアイテムファイルのフィールド NumberOfPlayItems の値が 1 だけインクリメントされ、当該プレイリストに対してプレイアイテムが 1 つ、追加されることが示される (図 1 1 参照)。

【 0 1 9 5 】

次のステップ S 5 7 で、ブロック blkPlayItem() におけるフィールド ClipInformationFileName、フィールド INTime およびフィールド OUTTime がそれぞれ設定され、クリップの記録に伴い追加されるブロック blkPlayItem() が作成される。フィールド ClipInformationFileName は、上述のステップ S 5 5 で作成されたクリップインフォメーションファイルのファイル名 "00002.clpi" が格納される。実際には、クリップインフォメーションファイルの拡張子は固定的とされているので、ピリオドの前の部分 "00002" が格納される。フィール

10

20

30

40

50

ドINTimeおよびフィールドOUTTimeは、対応するクリップA Vストリームファイル"00002.m2ts"に格納されるビデオストリームの先頭および終端の時間を示す情報であって、例えばクリップインフォメーションファイル"00002.clpi"内のブロックblkCPI()におけるブロックblkEPMaP()の情報に基づく。

【0196】

次のステップS58で、プレイリストファイル内のブロックblkPlayListMark()におけるフィールドNumberOfPlayListMarksの値が1だけインクリメントされ、それに伴いforループ文内に追加されたフィールドMarkTimeStampの値が、上述のステップS57でブロックblkPlayItem()におけるフィールドINTimeの値に設定される。すなわち、新たに記録されたクリップの先頭に、プレイリストマークが打たれる。

10

【0197】

このようにして、新たに記録されたクリップA Vストリームファイル"00002.m2ts"に対して、クリップインフォメーションファイル"00002.clpi"が作成されると共に、既存のプレイリストファイルが更新される。

【0198】

なお、上述したステップS53によるストリームバッファ14に溜め込まれたデータの記録媒体20への書き込み処理は、ステップS58の処理の後に行うようにしてもよい。

【0199】

記録が停止された状態から、UI部31に対する記録開始操作を行うことで、再びステップS50からの処理が開始され、新たなクリップA Vストリームファイルの記録媒体20への記録と、対応するクリップインフォメーションファイルの作成、作成されたクリップインフォメーションファイルを参照するプレイアイテムの生成、ならびに、生成されたプレイアイテムのプレイリストへの追加処理が同様に行われる。

20

【0200】

なお、一旦記録が停止されてから次の記録が開始される場合、上述したステップS54で作成したクリップインフォメーションファイルの記録媒体20への書き込みが終了するまで、次のクリップA Vストリームファイルの記録は、開始しないようにする。

【0201】

この場合、記録が停止されてから次の記録を開始できない可能性がある。これを避けるために、例えばクリップインフォメーションファイルを生成するためにワークメモリとして用いた揮発性メモリ17に、当該クリップインフォメーションファイルを保持しておき、例えばディスクイジェクト時や記録装置の電源OFF時など所定のタイミングで揮発性メモリ17に保持されているクリップインフォメーションファイルを記録媒体20に書き込むことが考えられる。

30

【0202】

なお、ここでは、記録媒体20上に既にクリップA V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およびプレイリストファイルが記録されている状態で、次のクリップA Vストリームファイルを記録する例について説明したが、図33のフローチャートを用いて説明した処理は、記録媒体20に最初のクリップA Vストリームファイルを記録する際にも、適用可能なものである。

40

【0203】

ここで、AVCHDフォーマットによれば、クリップA Vストリームファイルを参照するプレイアイテムが格納されるプレイリストファイルと、当該プレイリストファイルを呼び出すコマンドがムービーオブジェクトとして記述されたムービーオブジェクトファイルと、ムービーオブジェクトファイル中のムービーオブジェクトをタイトルとして呼び出すインデックスファイルとが必要とされる。記録媒体20に対して最初のクリップA Vストリームファイルを記録した際に、これらインデックスファイルおよびムービーオブジェクトファイルを自動的に作成されるようにできる。

【0204】

図34は、上述の図33の手順に従い作成された一例のファイル構造を示す。一例とし

50

て、記録媒体 20 に対して、最初のクリップ A V ストリームファイル "00001.m2ts" が記録され (ステップ S 50 ~ ステップ S 53)、対応するクリップインフォメーションファイル "00001.clpi" が作成される (ステップ S 54)。

【0205】

クリップ A V ストリームファイル "00001.m2ts" およびクリップインフォメーションファイル "00001.clpi" の記録および作成に応じて、次に、当該クリップインフォメーションファイル "00001.clpi" を参照するプレイアイテム # 1 の情報が生成されると共に、当該プレイアイテム # 1 を格納するプレイリストファイルが作成される (ステップ S 55 ~ ステップ S 58)。また、プレイリストファイルに対し、プレイアイテム # 1 のフィールド INTIME に対応する時刻にプレイリストマーク Mark#1 が打たれる (ステップ S 58)。

10

【0206】

なお、最初のクリップ A V ストリームファイル "00001.m2ts" の記録に伴い、プレイリスト # 1 を呼び出すムービーオブジェクト # 1 が記述されるムービーオブジェクトファイルと、ムービーオブジェクト # 1 をタイトル # 1 として選択および再生させるためのインデックスファイルを作成するようにできる。ムービーオブジェクトファイルおよびインデックスファイルは、テンプレートを予め作成しておき、それを更新するようにもできる。これらムービーオブジェクトファイルおよびインデックスファイルの作成については、この発明の主旨と関連性が薄いので、詳細な説明を省略する。

【0207】

次に記録開始操作がなされると、記録媒体 20 に対して、クリップ A V ストリームファイル "00002.m2ts" と、対応するクリップインフォメーションファイル "00002.clpi" とが追加して記録される (ステップ S 50 ~ ステップ S 53、ならびに、ステップ S 54)。クリップ A V ストリームファイル "00002.m2ts" およびクリップインフォメーションファイル "00002.clpi" の記録および作成に応じて、次に、当該クリップインフォメーションファイル "00002.clpi" を参照するプレイアイテム # 2 が生成される。この生成されたプレイアイテム # 2 は、既に作成されているプレイリストファイルに対して追加される。さらに、プレイリストファイルに対し、プレイアイテム # 2 のフィールド INTIME に対応する時刻にプレイリストマーク Mark#2 が打たれる (ステップ S 58)。

20

【0208】

このようにして、1 組の記録開始操作および記録停止操作により 1 個のクリップ A V ストリームファイルが記録媒体 20 に記録され、それに伴い、当該クリップ A V ストリームファイルに対応するクリップインフォメーションファイルが作成される。また、当該クリップインフォメーションファイルを参照するプレイアイテムが生成されて既存のプレイリストに対して追加され、当該プレイアイテムのフィールド INTIME で示される時刻にプレイリストマークが打たれる。これら一連の処理が記録開始操作および記録停止操作の組毎に、繰り返される。

30

【0209】

また、図 34 に例示されるファイル構造から分かるように、記録開始操作および記録停止操作の組の繰り返しにより記録される複数のクリップ A V ストリームファイルのそれぞれを参照する複数のプレイアイテムは、1 つのプレイリストファイルに、クリップ A V ストリームファイルの記録に伴い順次、追加して格納される。このプレイリストは、ムービーオブジェクトファイル内の 1 つのムービーオブジェクトから呼び出され、このムービーオブジェクトは、インデックスファイルから 1 つのタイトルとして選択および再生される。したがって、一連の記録開始操作および記録停止操作の組の繰り返しにより記録される複数のクリップ A V ストリームファイルは、インデックスファイル上で、1 つのタイトルとして管理されることになる。

40

【0210】

次に、この発明の実施の一形態の他の例について説明する。上述では、この発明が単体の記録装置に適用された例について説明した (図 32 参照)。これに対し、この実施の一形態の他の例では、この発明を、撮像素子と、被写体からの光を撮像素子に入射させる光

50

学系とを有し、撮像素子で撮像された撮像信号に基づきビデオデータを記録媒体に記録するようにした、ビデオカメラ装置に適用した。

【0211】

図35は、この発明の実施の一形態の他の例によるビデオカメラ装置100の一例の構成を示す。ビデオカメラ装置100において、記録系の構成は、図31を用いて説明した記録装置の構成を略そのまま適用できるので、図31と共通する部分には同一の符号を付し、詳細な説明を省略する。

【0212】

図35の構成において、カメラ部50は、映像信号に関する構成として、光学系51、撮像素子52、撮像信号処理部53、カメラ制御部54および表示部55を有し、音声信号に関する構成として、マイクロフォン(MIC)56および音声信号処理部57を有する。制御部30は、カメラ部50の各部との間で各種制御信号や情報のやりとりを行い、カメラ部50の動作を制御する。また、制御部50は、ユーザ操作に応じてUI部31から供給される制御信号に基づき、カメラ部50の動作を制御する。

【0213】

なお、ビデオカメラ装置100として構成される場合、記録開始操作および記録停止操作は、例えば、UI部31に設けられた単一の記録スイッチを用い、当該記録スイッチが押下される毎に記録開始および記録停止が交互に指示されるようになされるのが一般的である。また、このビデオカメラ装置100では、記録媒体20として、Blu-ray Discや記録可能なタイプのDVDといった、ディスク記録媒体を適用するものとする。

【0214】

カメラ部50において、光学系51は、被写体からの光を撮像素子52に導くためのレンズ系、絞り調整機構、フォーカス調整機構、ズーム機構、シャッタ機構などを備える。絞り調整機構、フォーカス調整機構、ズーム機構およびシャッタ機構の動作は、制御部30から供給される制御信号に基づき、カメラ制御部54により制御される。

【0215】

撮像素子52は、例えばCCD(Charge Coupled Device)からなり、光学系51を介して照射された光を光電変換により電気信号に変換し、所定の信号処理を施し撮像信号として出力する。撮像信号処理部53は、撮像素子から出力された撮像信号に対して所定の信号処理を施し、ベースバンドのデジタルビデオデータとして出力する。

【0216】

例えば撮像信号処理部53は、撮像素子52から出力された撮像信号に対して、CDS(Correlated Double Sampling)回路により画像情報を有する信号だけをンプリングすると共に、ノイズを除去し、AGC(Auto Gain Control)回路によりゲインを調整する。そして、A/D変換によりデジタル信号に変換する。また、撮像信号処理部53は、このデジタル信号に対して検波系の信号処理を施し、R(赤色)、G(緑色)およびB(青色)各色の成分を取り出し、補正やホワイトバランス補正などの処理を行い、最終的に1本のベースバンドのデジタルビデオデータとして出力する。

【0217】

また、撮像信号処理部53は、撮像素子52から出力された撮像信号の情報を制御部30に送る。制御部30は、この情報に基づき光学系51を制御するための制御信号を生成し、カメラ制御部54に供給する。カメラ制御部54は、この制御信号に基づきフォーカス調整機構や絞り調整機構などの制御を行う。

【0218】

さらに、撮像信号処理部53は、撮像素子52から出力された撮像信号に基づき、例えばLCD(Liquid Crystal Display)を表示素子として用いた表示部55に映出させる映像信号を生成する。

【0219】

一方、マイクロフォン56は、周囲の音声を収音して電気信号に変換して出力する。マ

10

20

30

40

50

イクロフォン 56 から出力された音声信号は、音声信号処理部 57 に供給される。音声信号処理部 57 は、供給された音声信号を、リミッタを介してから A/D 変換を施してデジタルオーディオデータとし、ノイズ除去や音質補正など所定の音声信号処理を施してベースバンドのデジタルオーディオデータとして出力する。

【0220】

カメラ部 50 の撮像信号処理部 53 から出力されたベースバンドのデジタルビデオデータは、記録部 10 の端子 40 に供給される。また、音声信号処理部 57 から出力されたベースバンドのデジタルオーディオデータは、記録部 10 の端子 41 に供給される。

【0221】

記録停止状態から UI 部 31 に設けられた記録スイッチが押下されると、記録開始を指示する制御信号が UI 部 31 から制御部 30 に供給され、制御部 30 の制御に基づきカメラ部 50 から出力されたベースバンドのデジタルビデオ信号およびデジタルオーディオデータの記録媒体 20 への記録が開始される。

10

【0222】

すなわち、既に説明したように、制御部 30 の制御に基づきビデオエンコーダ 11 およびオーディオエンコーダ 12 の動作が開始され、ビデオデータおよびオーディオデータがそれぞれビデオエンコーダ 11 およびオーディオエンコーダ 12 で圧縮符号化され、マルチプレクサ 13 で所定にパケット化され多重化されて AV ストリームデータとされる。AV ストリームデータは、ストリームバッファ 14 を介して、記録制御部 15 に供給され、クリップ AV ストリームファイルとして記録媒体 20 に記録される。

20

【0223】

UI 部 31 の記録スイッチが押下されると、記録が停止され、クリップインフォメーションファイルの作成や、プレイリストファイルの更新が行われる。管理情報処理部 16 は、マルチプレクサ 13 および記録制御部 15 からの情報に基づき、記録媒体 20 に記録されるクリップ AV ストリームファイルに対応するクリップインフォメーションファイルを作成する。また、管理情報処理部 16 は、当該クリップインフォメーションファイルを参照するプレイアイテムを生成し、既にプレイリストが存在する場合には、生成されたプレイアイテムを当該プレイリストに対して追加すると共に、プレイリストに対してプレイリストマークを打つ。

【0224】

30

この状態からもう一度記録スイッチが押下されると、再び記録開始が指示され、新たなクリップ AV ストリームファイルの記録媒体 20 への記録が開始されると共に、対応するクリップインフォメーションファイルの作成や、当該クリップインフォメーションファイルを参照するプレイアイテムの生成、生成されたプレイアイテムのプレイリストへの追加などの処理が行われる。クリップ AV ストリームファイル間の再生は、第 1 のシームレス接続により接続される。

【0225】

なお、記録停止に伴い作成されるクリップインフォメーションファイルを、作成直後から記録媒体 20 に書き込む方式の場合は、直前に作成されたクリップインフォメーションファイルの記録媒体 20 への書き込みが終了するまで、次の記録が開始されないように制御される。

40

【0226】

この実施の一形態の他の例のように、この発明をビデオカメラ装置 100 に適用する場合、1 の記録媒体 20 に対して、プレイリストファイルは 1 ファイルのみが存在できるように制限することが考えられる。すなわち、編集処理などがなされない限り、1 の記録媒体 20 上には、当該プレイリストの再生を行うための 1 のタイトルしか存在しないようにする。こうすることで、記録媒体 20 としてディスクを用いたビデオカメラ装置 100 を、恰も従来の磁気テープを記録媒体に用いたビデオカメラ装置であるかのように扱うことができる。

【0227】

50

すなわち、この実施の一形態の他の例によれば、それぞれ個別に記録されるクリップ A V ストリームファイルが 1 のプレイリストにより管理され、各クリップ A V ストリームファイル間の再生が第 1 のシームレス接続で接続されるため、記録媒体 20 に記録された全てのクリップ A V ストリームファイルを、磁気テープを再生する場合のように、連続的に再生することができる。また、プレイリストに対し、それぞれ個別に記録されるクリップ A V ストリームファイルの先頭に対応する位置にプレイリストマークが打たれているため、クリップ単位での検索が容易である。

【0228】

なお、上述では、図 31 に示す記録装置や図 35 に示すビデオカメラ装置 100 の記録部 10 がハードウェア的に構成されるように説明したが、これはこの例に限定されない。すなわち、記録部 10 は、ソフトウェアとして構成することも可能である。この場合、ソフトウェアは、例えば制御部 30 が有する図示されない ROM に予め記憶される。これに限らず、記録部 10 を、パーソナルコンピュータなどのコンピュータ装置上に構成することも可能である。この場合には、記録部 10 をコンピュータ装置に実行させるソフトウェアは、CD-ROM や DVD-ROM といった記録媒体に記録されて提供される。コンピュータ装置がネットワーク接続可能な場合、インターネットなどのネットワークを介して当該ソフトウェアを提供することも可能である。

【図面の簡単な説明】

【0229】

【図 1】この発明に適用可能な A V C H D フォーマットに規定されるデータモデルを概略的に示す略線図である。

【図 2】インデックステーブルを説明するための略線図である。

【図 3】クリップ A V ストリーム、クリップ情報、クリップ、プレイアイテムおよびプレイリストの関係を示す U M L 図である。

【図 4】複数のプレイリストから同一のクリップを参照する方法を説明するための略線図である。

【図 5】記録媒体に記録されるファイルの管理構造を説明するための略線図である。

【図 6】ファイル "index.bdmv" の一例の構造を表すシンタクスを示す略線図である。

【図 7】ブロック blkIndexes() の一例の構造を表すシンタクスを示す略線図である。

【図 8】ファイル "MovieObject.bdmv" の一例の構造を表すシンタクスを示す略線図である

。

【図 9】ブロック blkMovieObjects() の一例の構造を表すシンタクスを示す略線図である

。

【図 10】プレイリストファイル "xxxxx.mpls" の一例の構造を表すシンタクスを示す略線図である。

【図 11】ブロック blkPlayList() の一例の構造を表すシンタクスを示す略線図である。

【図 12】ブロック blkPlayItem() の一例の構造を表すシンタクスを示す略線図である。

【図 13】第 1 および第 2 のシームレス接続を説明するための略線図である。

【図 14】ブロック blkPlayListMark() の一例の構造を表すシンタクスを示す略線図である。

【図 15】クリップインフォメーションファイルの一例の構造を表すシンタクスを示す略線図である。

【図 16】ブロック blkClipInfo() の一例の構造を表すシンタクスを示す略線図である。

【図 17】ブロック blkSequenceInfo() の一例の構造を表すシンタクスを示す略線図である。

【図 18】ブロック blkProgramInfo() の一例の構造を表すシンタクスを示す略線図である

。

【図 19】ブロック blkCPI() の一例の構造を表すシンタクスを示す略線図である。

【図 20】ブロック blkEPMAP() の一例の構造を表すシンタクスを示す略線図である。

【図 21】ブロック blkEPMAPForOneStreamPID(EP_stream_type, Nc, Nf) の一例の構造を

10

20

30

40

50

表すシンタクスを示す略線図である。

【図 2 2】エン트리PTSEPCoarseおよびエン트리PTSEPFineの一例のフォーマットについて示す略線図である。

【図 2 3】エン트리SPNEPCoarseおよびエン트리SPNEPFineの一例のフォーマットについて示す略線図である。

【図 2 4】ブロックblkExtensionData()の一例の構造を表すシンタクスを示す略線図である。

【図 2 5】ブロックblkExtensionData()における各データの参照関係を模式的に示す略線図である。

【図 2 6】ブロックblkExtensionData()にデータを書き込む際の一例の処理を示すフローチャートである。

10

【図 2 7】ブロックblkExtensionData()から拡張データを読み出す際の一例の処理を示すフローチャートである。

【図 2 8】ファイル"index.bdmv"内のフィールドblkExtensionData()におけるブロックDataBlock()の一例の構造を表すシンタクスを示す略線図である。

【図 2 9】ブロックblkTableOfPlayList()の一例の構造を表すシンタクスを示す略線図である。

【図 3 0】仮想プレーヤの動作を概略的に示すフローチャートである。

【図 3 1】仮想プレーヤの動作を概略的に示す略線図である。

【図 3 2】この発明の実施の一形態に適用可能な記録装置の一例の構成を概略的に示すブロック図である。

20

【図 3 3】この発明の実施の一形態によるクリップの一例の記録方法を示すフローチャートである。

【図 3 4】この発明の実施の一形態の方法により記録された一例のファイル構造を示す略線図である。

【図 3 5】この発明の実施の一形態の他の例によるビデオカメラ装置の一例の構成を示す。

【符号の説明】

【 0 2 3 0 】

1 0 記録部

30

1 1 ビデオエンコーダ

1 2 オーディオエンコーダ

1 3 マルチプレクサ

1 4 ストリームバッファ

1 5 記録制御部

1 6 管理情報処理部

1 7 揮発性メモリ

1 8 不揮発性メモリ

2 0 記録媒体

3 0 制御部

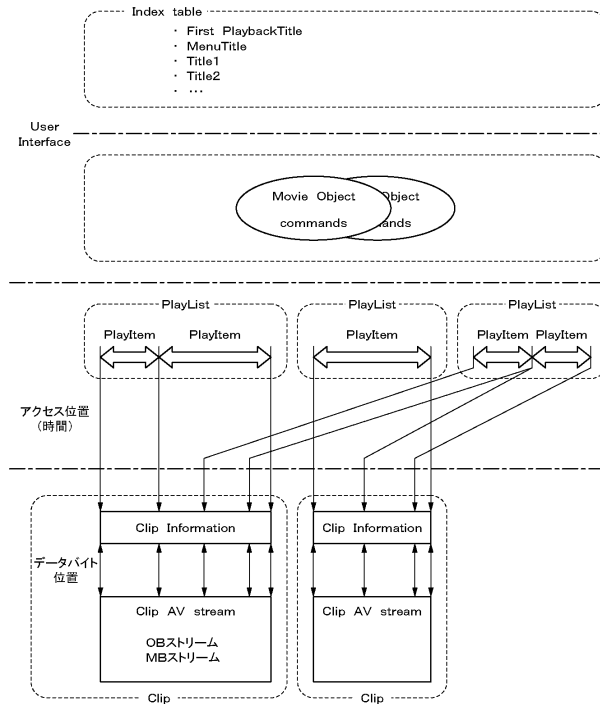
40

3 1 ユーザインターフェイス部

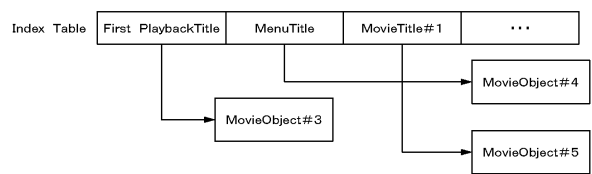
5 0 カメラ部

1 0 0 ビデオカメラ装置

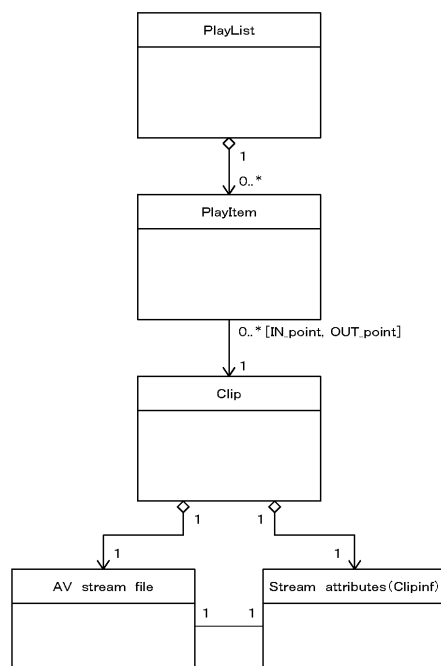
【図 1】



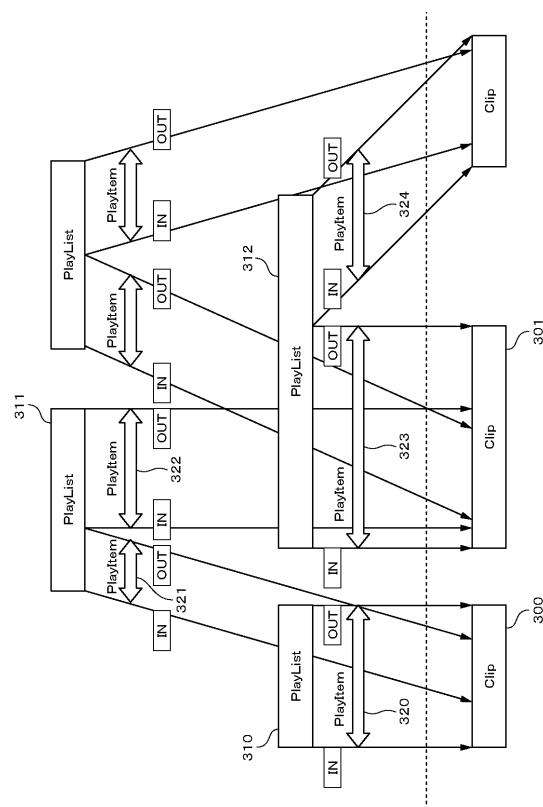
【図 2】



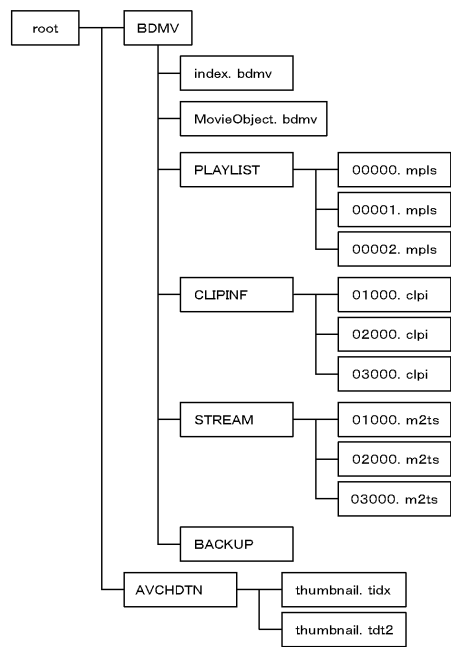
【図 3】



【図 4】



【図 5】



【図 6】

シンタクス	データ長 (ビット)	ニーモニック
Index table file {		
TypeIndicator	8 * 4	bslbf
TypeIndicator2	8 * 4	bslbf
IndexesStartAddress	32	uimbsf
ExtensionDataStartAddress	32	uimbsf
reserved	192	bslbf
blkAppInfoBDMV()		
for (i=0; i<N1; i++) {		
padding_word	16	bslbf
}		
blkIndexes()		
for (i=0; i<N2; i++) {		
padding_word	16	bslbf
}		
blkExtensionData()		
for (i=0; i<N3; i++) {		
padding_word	16	bslbf
}		
}		

【図 7】

シンタクス	データ長 (ビット)	ニーモニック
blkIndexes() {		
Length	32	uimbsf
FirstPlaybackTitle() {		
reserved	1	bslbf
'1'	1	bslbf
reserved	31	bslbf
'1'	1	bslbf
reserved	14	bslbf
FirstPlaybackTitleMobjIDRef	16	uimbsf
reserved	32	bslbf
}		
MenuTitle() {		
reserved	1	bslbf
'1'	1	bslbf
reserved	31	bslbf
'1'	1	bslbf
reserved	14	bslbf
MenuTitleMobjIDRef	16	uimbsf
reserved	32	bslbf
}		
NumberOfTitles	16	uimbsf
for (title_id=0; title_id < NumberOfTitles; title_id++) {		
MovieTitle[title_id]() {		
reserved	1	bslbf
'1'	1	bslbf
reserved	46	bslbf
MovieTitleMobjIDRef[title_id]	16	uimbsf
reserved	32	bslbf
}		
}		
}		

【図 8】

シンタクス	データ長 (ビット)	ニーモニック
MovieObject file {		
TypeIndicator	8 * 4	bslbf
TypeIndicator2	8 * 4	bslbf
ExtensionDataStartAddress	32	uimbsf
reserved	224	bslbf
blkMovieObjects()		
for (i=0; i<N1; i++) {		
padding_word	16	bslbf
}		
blkExtensionData()		
for (i=0; i<N2; i++) {		
padding_word	16	bslbf
}		
}		

【図 9】

シンタクス	データ長 (ビット)	ニーモニック
blkMovieObjects() {		
Length	32	uimbsf
reserved	32	bslbf
NumberOfMobjs	16	uimbsf
for (mobj_id=0; mobj_id<NumberOfMobjs; mobj_id++) {		
MovieObject[mobj_id]() {		
TerminalInfo() {		
'1'	1	bslbf
reserved	15	bslbf
}		
NumberOfNavigationCommands[mobj_id]	16	uimbsf
for (command_id=0; command_id<NumberOfNavigationCommands[mobj_id]; command_id++){		
NavigationCommand[mobj_id][command_id]	96	bslbf
}		
}		
}		
}		

【図 10】

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
}		
}		

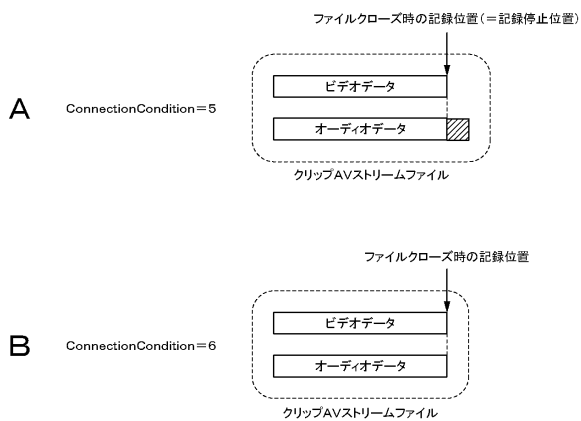
【図 11】

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 12】

シンタクス	データ長 (ビット)	ニーモニック
blkPlayItem() {		
Length	16	uimsbf
ClipInformationFileName	8 * 5	bslbf
ClipCodeIdentifier	8 * 4	bslbf
reserved	12	bslbf
ConnectionCondition	4	bslbf
RefToSTCID	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
}		
blkSTNTTable()		
}		

【図 13】



【図 14】

シンタクス	データ長 (ビット)	ニーモニック
blkPlayListMark() {		
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 15】

シンタクス	データ長 (ビット)	ニーモニック
Clip information file {		
TypeIndicator	8 * 4	bslbf
TypeIndicator2	8 * 4	bslbf
SequenceInfoStartAddress	32	uimsbf
ProgramInfoStartAddress	32	uimsbf
CPIStartAddress	32	uimsbf
ClipMarkStartAddress	32	uimsbf
ExtensionDataStartAddress	32	uimsbf
reserved	96	bslbf
blkClipInfo() for (i=0; i<N1; i++) {		
padding_word	16	bslbf
}		
blkSequenceInfo() for (i=0; i<N2; i++) {		
padding_word	16	bslbf
}		
blkProgramInfo() for (i=0; i<N3; i++) {		
padding_word	16	bslbf
}		
blkCPI() for (i=0; i<N4; i++) {		
padding_word	16	bslbf
}		
blkClipMark() for (i=0; i<N5; i++) {		
padding_word	16	bslbf
}		
blkExtensionData() for (i=0; i<N6; i++) {		
padding_word	16	bslbf
}		
}		

【図 16】

シンタクス	データ長 (ビット)	ニーモニック
blkClipInfo() {		
Length	32	uimsbf
reserved	16	bslbf
ClipStreamType	8	bslbf
ApplicationType	8	bslbf
reserved	31	bslbf
IsCC5	1	bslbf
TSRecordingRate	32	uimsbf
NumberOfSourcePackets	32	uimsbf
reserved	1024	bslbf
TSTypeInfoBlock()		
if (IsCC5 == 1 _b) {		
reserved	8	bslbf
FollowingClipStreamType	8	bslbf
reserved	32	bslbf
FollowingClipInformationFileName	8 * 5	bslbf
ClipCodeIdentifier	8 * 4	bslbf
reserved	8	bslbf
}		
}		

【図 17】

シンタクス	データ長 (ビット)	ニーモニック
blkSequenceInfo() {		
Length	32	uimsbf
reserved	15	bslbf
'1'	1	bslbf
SPNATCStart	32	uimsbf
NumberOfSTCSequences	8	uimsbf
reserved	8	uimsbf
for (stc_id=0; stc_id<NumberOfSTCSequences; stc_id++) {		
PCRPID[stc_id]	16	uimsbf
SPNSTCStart[stc_id]	32	uimsbf
PresentationStartTime[stc_id]	32	uimsbf
PresentationEndTime[stc_id]	32	uimsbf
}		
}		

【図 20】

シンタクス	データ長 (ビット)	ニーモニック
blkEPMAP(){		
reserved	8	bslbf
NumberOfStreamPIDEntries	8	uimsbf
for (k=0; k<NumberOfStreamPIDEntries; k++) {		
StreamPID[k]	16	bslbf
reserved	10	bslbf
EPStreamType[k]	4	bslbf
NumberOfEPCoarseEntries[k]	16	uimsbf
NumberOfEPFineEntries[k]	18	uimsbf
EPMapForOneStreamPIDStartAddress[k]	32	uimsbf
}		
for (i=0; i<X; i++) {		
padding_word	16	bslbf
}		
for (k=0; k<NumberOfStreamPIDEntries; k++) {		
blkEPMAPForOneStreamPID (EPStreamType[k], NumberOfEPCoarseEntries[k], NumberOfEPFineEntries[k])		
for (i=0; i<Y[k]; i++) {		
padding_word	16	bslbf
}		
}		
}		

【図 18】

シンタクス	データ長 (ビット)	ニーモニック
blkProgramInfo() {		
Length	32	uimsbf
reserved	15	bslbf
'1'	1	bslbf
SPNProgramSequenceStart	32	uimsbf
ProgramMapPID	16	uimsbf
NumberOfStreamsInPS	8	uimsbf
reserved	8	bslbf
for (stream_index=0; stream_index<NumberOfStreamsInPS; stream_index++) {		
StreamPID[stream_index]	16	uimsbf
blkStreamCodingInfo (stream_index)		
}		
}		

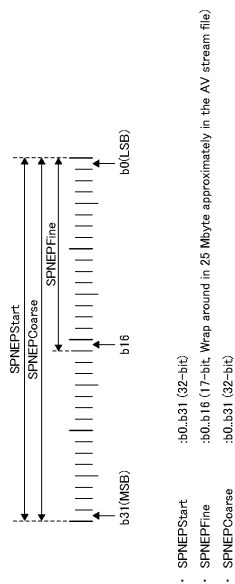
【図 19】

シンタクス	データ長 (ビット)	ニーモニック
blkCPI() {		
Length	32	uimsbf
if(Length !=0) {		
reserved	12	bslbf
CPITYPE	4	bslbf
blkEPMAP()		
}		
}		

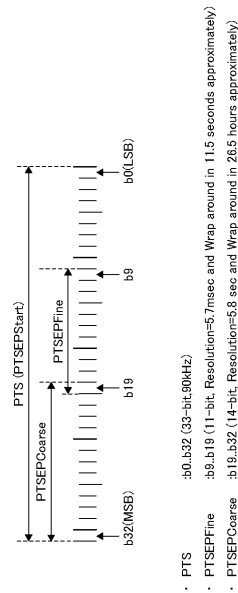
【図 21】

シンタクス	データ長 (ビット)	ニーモニック
blkEPMAPForOneStreamPID(EP_stream_type, Nc, Nf) {		
EPFineTableStartAddress	32	uimsbf
for (i=0; i<Nc; i++) {		
// EP coarse table		
RefToEPFineID[i]	18	uimsbf
PTSEPCoarse[i]	14	uimsbf
SPNEPCoarse[i]	32	uimsbf
}		
for (i=0; i<X; i++) {		
padding_word	16	bslbf
}		
for (EP_fine_id=0; EP_fine_id < Nf, EP_fine_id++) {		
// EP fine table		
ReservedEPFine [EP_fine_id]	1	bslbf
IEndPositionOffset [EP_fine_id]	3	bslbf
PTSEPFine [EP_fine_id]	11	uimsbf
SPNEPFine [EP_fine_id]	17	uimsbf
}		
}		

【図 2 2】



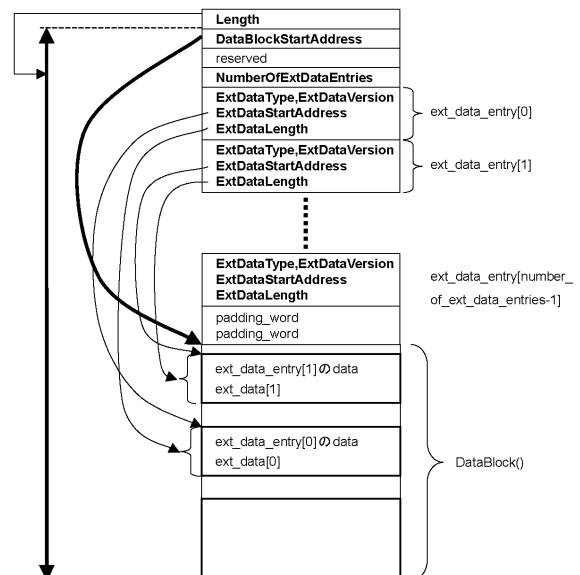
【図 2 3】



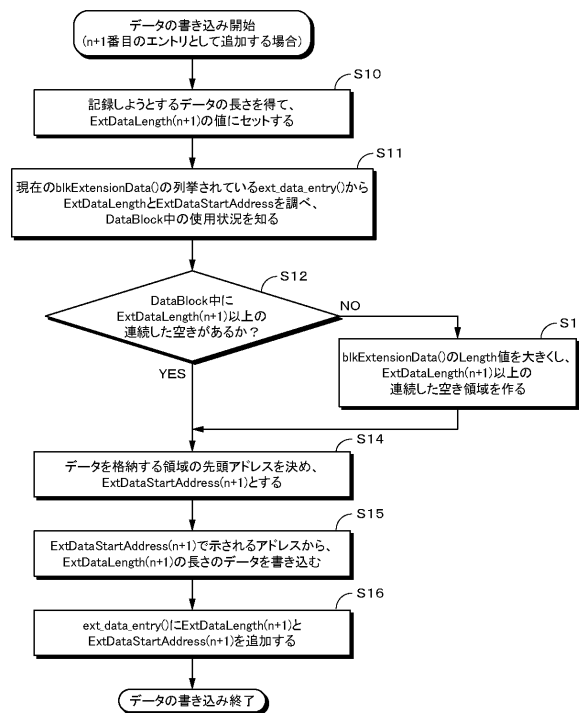
【図 2 4】

シンタックス	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	
NumberOfExtDataEntries	8	uimsbf
for (i=0; i<NumberOfExtDataEntries; i++) {		
ext_data_entry() {		
ExtDataType	16	uimsbf
ExtDataVersion	16	uimsbf
ExtDataStartAddress	32	uimsbf
ExtDataLength	32	uimsbf
}		
}		
for (i=0; i<L1; i++) {		
padding_word	16	bslbf
padding_word	16	bslbf
}		
DataBlock()	32 + 8 * (Length - DataBlockStartAddress)	
}		
}		

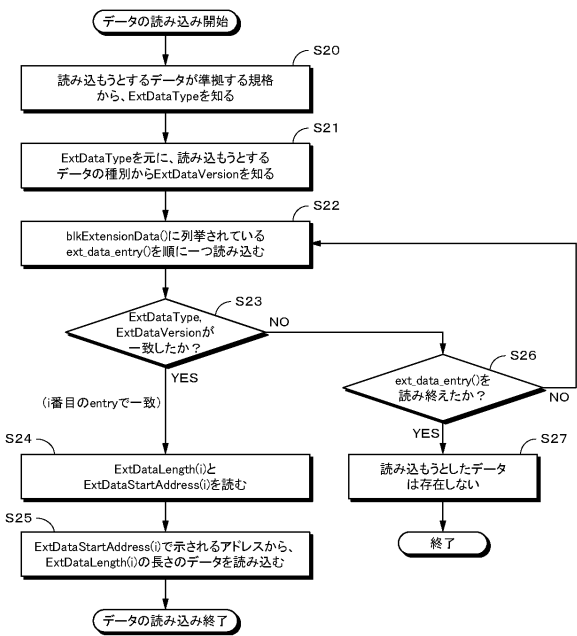
【図 2 5】



【図 26】



【図 27】



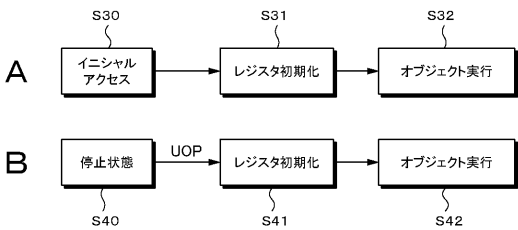
【図 28】

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

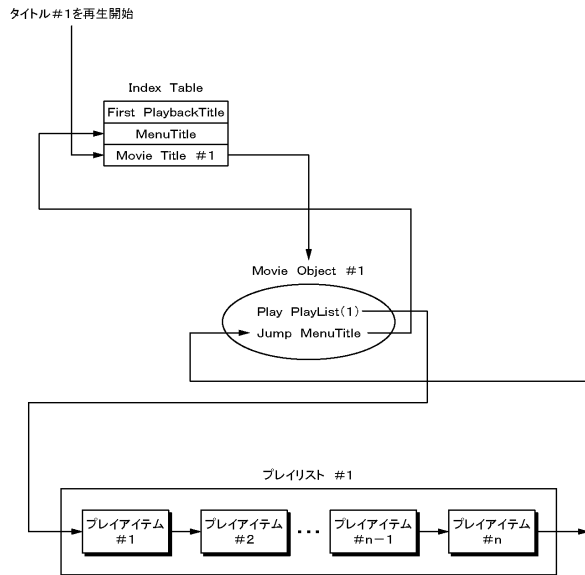
【図 29】

シンタクス	データ長 (ビット)	ニーモニック
blkTableOfPlayLists() {		
Length	32	uimbsf
blkFirstPlaybackTitlePlayLists()		
blkMenuTitlePlayLists()		
NumberOfTitlePlaylistPair	16	bslbf
for(i=0; i< NumberOfTitlePlaylistPair;i++){		
blkMovieTitlePlaylistPair() {		
PlaylistFileName	8 * 5	bslbf
reserved	6	bslbf
PlaylistAttribute	2	uimbsf
reserved	16	bslbf
RefToTitleId	16	uimbsf
}		
}		
}		

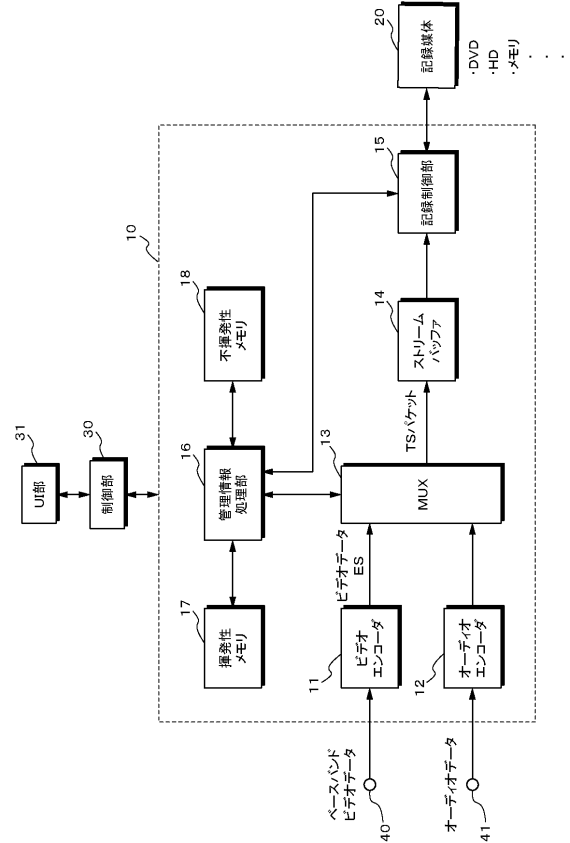
【図 30】



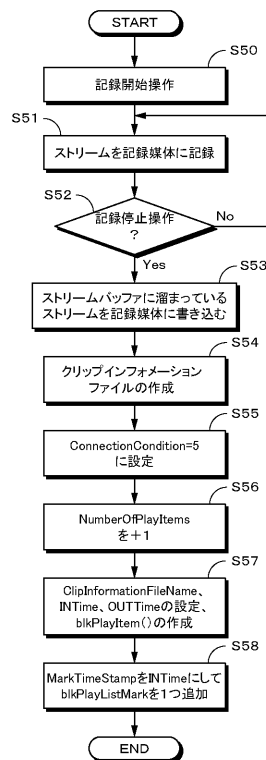
【 図 3 1 】



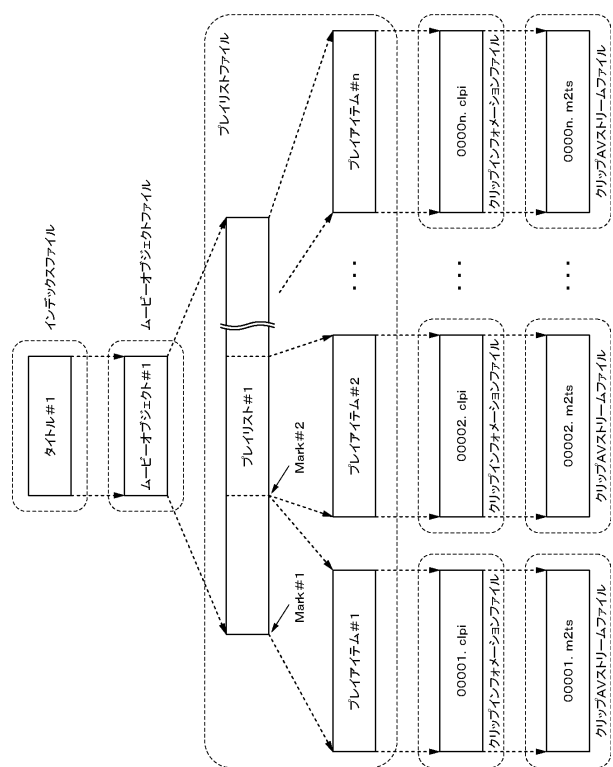
【 図 3 2 】



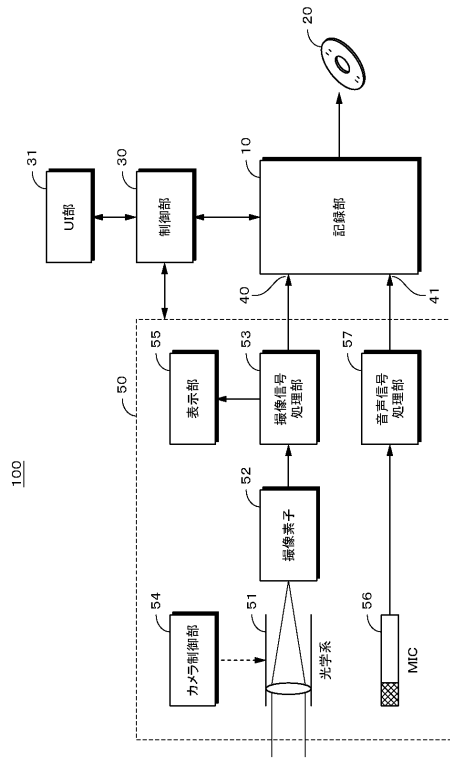
【 図 3 3 】



【 図 3 4 】



【図 35】



フロントページの続き

(51)Int.Cl. F I
G 1 1 B 27/10 (2006.01) G 1 1 B 20/12
G 1 1 B 27/10 A

(72)発明者 前 篤
東京都品川区北品川6丁目7番35号 ソニー株式会社内
(72)発明者 前田 哲裕
東京都品川区北品川6丁目7番35号 ソニー株式会社内

審査官 宮下 誠

(56)参考文献 特開2005-303943(JP,A)
特開2002-290917(JP,A)
特開2003-230104(JP,A)
特開2005-353212(JP,A)
特開2005-216382(JP,A)
特開平11-317014(JP,A)

(58)調査した分野(Int.Cl., DB名)
G 1 1 B 27 / 00
G 1 1 B 20 / 10
H 0 4 N 5 / 7 6
H 0 4 N 5 / 9 1