

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2012-170112

(P2012-170112A)

(43) 公開日 平成24年9月6日(2012.9.6)

(51) Int.Cl.	F I	テーマコード (参考)
HO4N 5/92 (2006.01)	HO4N 5/92 C	5C052
HO4N 5/91 (2006.01)	HO4N 5/91 Z	5C053
HO4N 5/85 (2006.01)	HO4N 5/85 Z	5D044
G11B 20/10 (2006.01)	G11B 20/10 311	
G11B 20/12 (2006.01)	G11B 20/12	

審査請求 有 請求項の数 3 O L (全 59 頁) 最終頁に続く

(21) 出願番号 特願2012-88609 (P2012-88609)
 (22) 出願日 平成24年4月9日 (2012.4.9)
 (62) 分割の表示 特願2009-8702 (P2009-8702)
 の分割
 原出願日 平成15年4月25日 (2003.4.25)

(特許庁注：以下のものは登録商標)

1. B l u - r a y

(71) 出願人 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
 (74) 代理人 100123973
 弁理士 杉浦 拓真
 (72) 発明者 浜田 俊也
 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 加藤 元樹
 東京都品川区北品川6丁目7番35号 ソ
 ニー株式会社内
 Fターム(参考) 5C052 AA02 CC11 DD04

最終頁に続く

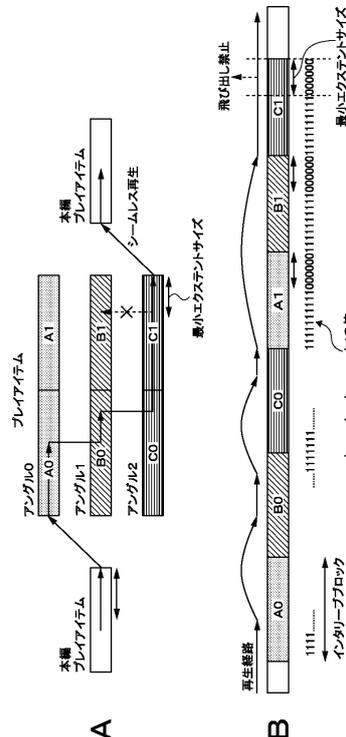
(54) 【発明の名称】 データ格納方法

(57) 【要約】

【課題】大容量の記録済み媒体において自由度が高く表現力の豊かな高いユーザインターフェイスを実現する。

【解決手段】プレイアイテムが複数のアングルで再生できるマルチアングル構造であるか否かを示すフラグを設ける。また、デコード単位の先頭にアングル切り替え可能点であるか否かを示すフラグを設ける。シームレスのマルチアングル構造において、インターリーブユニットを増やさなくてもアングル切り換えが可能となる。ノンシームレスのマルチアングル構造では、このフラグを用いて各アングルの末尾側の所定領域をアングル切り替え不可とすることで、マルチアングルから脱したときの再生の不連続を無くす。また、サブプレイアイテムをメインパスと非同期に再生可能か否かを示すフラグを設けることで、オーディオのみのサブプレイアイテムをBGMとして用いることができる。

【選択図】 図64



【特許請求の範囲】**【請求項 1】**

再生データを記録媒体に記録する記録方法であって、

1 または複数のフレームを単位としてエンコードされているビデオストリームと、該エンコード単位毎に設けられた、該ビデオストリームのアングルを該エンコード単位の先頭で切り替え可能か否かを示す識別情報とを含み、

上記識別情報に基づく位置で、上記アングル毎にインタリーブされて記録された上記ビデオストリームの再生位置を変更して該アングルを構成するビデオストリームを読み出すことが可能であるデータ構造を有するデータを記録媒体に記録する記録方法。

【請求項 2】

請求項 1 に記載の記録方法であって、

上記識別情報が上記アングルの終端側の所定領域に設けられている記録方法。

【請求項 3】

請求項 2 に記載の記録方法であって、

上記所定領域の大きさは、第 1 の領域から第 2 の領域への最大アクセス時間と、上記ビデオストリームの読み出し速度と該ビデオストリームの再生速度との差分とに基づく記録方法。

【請求項 4】

再生データが記録される記録媒体の製造方法であって、

1 または複数のフレームを単位としてエンコードされているビデオストリームと、該エンコード単位毎に設けられた、該ビデオストリームのアングルを該エンコード単位の先頭で切り替え可能か否かを示す識別情報とを含み、

上記識別情報に基づく位置で、上記アングル毎にインタリーブされて記録された上記ビデオストリームの再生位置を変更して該アングルを構成するビデオストリームを読み出すことが可能であるデータ構造を有するデータを記録媒体に記録する工程を含む記録媒体の製造方法。

【発明の詳細な説明】**【技術分野】****【0001】**

この発明は、ブルーレイディスク(Blu-ray Disc)といった大容量の記録媒体に記録されたプログラムに対するユーザによるインタラクティブな操作を可能とする記録方法および記録媒体の製造方法に関する。

【背景技術】**【0002】**

近年、記録可能で記録再生装置から取り外し可能なディスク型記録媒体の規格として、Blu-ray Disc (ブルーレイディスク)規格が提案されている。Blu-ray Disc 規格では、記録媒体として直径 12 cm、カバー層 0.1 mm のディスクを用い、光学系として波長 405 nm の青紫色レーザ、開口数 0.85 の対物レンズを用いて、最大で 27 GB (ギガバイト) の記録容量を実現している。これにより、日本の BS デジタルハイビジョン放送を、画質を劣化させることなく 2 時間以上記録することが可能である。

【0003】

この記録可能光ディスクに記録する AV (Audio/Video) 信号のソース (供給源) としては、従来からの、例えばアナログテレビジョン放送によるアナログ信号によるものと、例えば BS デジタル放送をはじめとするデジタルテレビジョン放送によるデジタル信号によるものが想定されている。Blu-ray Disc 規格では、これらの放送による AV 信号を記録する方法を定めた規格は、既に作られている。

【0004】

一方で、現状の Blu-ray Disc の派生規格として、映画や音楽などが予め

10

20

30

40

50

記録された、再生専用の記録媒体を開発する動きが進んでいる。映画や音楽を記録するためのディスク状記録媒体としては、既にDVD(Digital Versatile Disc)が広く普及しているが、このBlu-ray Discの企画に基づいた再生専用光ディスクは、Blu-ray Discの大容量および高速な転送速度などを活かし、ハイビジョン映像を高画質なままで2時間以上収録できる点が、既存のDVDとは大きく異なり、優位である。

【発明の概要】

【発明が解決しようとする課題】

【0005】

ところで、現状のBlu-ray Discの規格では、ディスクに記録されている映像コンテンツの一覧を画面表示する方法や、その一覧表上にカーソルを表示させ、再生したい映像コンテンツをユーザに選択させるなどといったユーザインターフェイスに関する機能が定められていない。これらの機能は、Blu-ray Discに対する記録再生を行う記録再生装置本体によって実現されている。そのため、同一の記録媒体を再生した場合でも、再生に用いた記録再生装置によってコンテンツ一覧画面のレイアウトが異なってしまう、ユーザインターフェイスにも差が生じ、必ずしもユーザにとって使い易いものではない。再生専用ディスクとしては、再生機器によらず、ディスク(コンテンツ)制作者が意図した通りのメニュー画面などが表示され、意図通りのユーザインターフェイスが実現される必要がある。

10

【0006】

さらに、映像コンテンツの再生中に選択画面が表示され、ユーザの選択によってストーリーが分岐していくマルチストーリーの機能は、一般にインタラクティブ機能とも呼ばれる。このインタラクティブ機能を実現するためには、ディスク制作者が再生順序や分岐を定めたシナリオを作り、そのシナリオをプログラム言語、スクリプト言語等を使って記述し、ディスクに記録しておく必要がある。再生装置側では、そのプログラムを読み込み、実行することで、制作者の意図に従った映像コンテンツの再生や、分岐のための選択画面提示を実現することになる。

20

【0007】

このように、現状のBlu-ray Disc規格(Blu-ray Disc Rewritable Format Ver1.0)では、この制作者の意図通りのユーザインターフェイスを実現するための、メニュー画面や分岐選択画面の構成方法、ユーザ入力に対する処理を記述する方法が定められていない。そのため、現状では、Blu-ray Discを用いて、制作者が意図したシナリオ通りの再生を、再生装置の製造メーカーや機種に左右されることなく互換性を持たせた形で実現することが難しいという問題点があった。

30

【0008】

また、再生専用ディスクにおいては、同一の被写体を複数のカメラで撮影し、ユーザが好みのアングルでの視聴を選択できるマルチアングル機能が制作者側より求められており、これを実現できる仕組みを用意する必要がある。

【0009】

一方、従来から、例えばDVD(Digital Versatile Disc)の規格においては、上述のようなインタラクティブな機能が既の実現されていた。例えば、DVDビデオにより動画を再生中に、リモートコントロールコマンドなどを用いてメニュー画面を呼び出し、例えばメニュー画面上に配置されたボタンを選択するなどして、再生場面を変更するなどの処理が可能であった。また、マルチアングル機能も実現されていた。

40

【0010】

このように、DVDといった従来の記録媒体で実現されていたようなユーザインターフェイスやインタラクティブな機能を、Blu-ray Discといった大容量の記録媒体を再生専用ディスクとして用いた場合にも実現できることが求められている。

【0011】

したがって、この発明の目的は、大容量の記録済み媒体において自由度の高いユーザインターフェイスを実現するための記録媒体を提供することにある。

50

【 0 0 1 2 】

また、この発明の他の目的は、大容量の記録済み媒体において、より表現豊かなユーザインターフェイスを実現する記録媒体を提供することにある。

【課題を解決するための手段】

【 0 0 1 3 】

この発明は、上述した課題を解決するために、再生データを記録媒体に記録する記録方法であって、データ構造は、1または複数のフレームを単位としてエンコードされているビデオストリームと、エンコード単位毎に設けられた、ビデオストリームのアングルをエンコード単位の先頭で切り替え可能か否かを示す識別情報とを含み、識別情報に基づく位置で、アングル毎にインタリーブされて記録されたビデオストリームの再生位置を変更してアングルを構成するビデオストリームを読み出すことが可能であるデータ構造を有するデータを記録媒体に記録する記録方法である。

10

この発明は、再生データが記録される記録媒体の製造方法であって、1または複数のフレームを単位としてエンコードされているビデオストリームと、該エンコード単位毎に設けられた、該ビデオストリームのアングルを該エンコード単位の先頭で切り替え可能か否かを示す識別情報とを含み、識別情報に基づく位置で、アングル毎にインタリーブされて記録されたビデオストリームの再生位置を変更して該アングルを構成するビデオストリームを読み出すことが可能であるデータ構造を有するデータを記録媒体に記録する工程を含む記録媒体の製造方法である。

20

【発明の効果】

【 0 0 1 4 】

上述したように、この発明は、記録媒体に、ビデオストリームと、ビデオストリームの再生単位毎に設けられた、再生単位が複数のアングルで再生できるものであるか否かを示す第1のフラグと、再生単位が含むアングル数と、アングルそれぞれのビデオストリーム上での位置を示す位置情報とが記録され、位置情報に基づきアングルを構成するビデオストリームを読み出すことができるようにしているため、再生単位のビデオストリームを再生中にアングルを切り替えることができる効果がある。

【 0 0 1 5 】

また、この発明は、主として再生されるメインストリームと、メインストリームに対して従としてメインストリームと共に再生されるサブストリームと、サブストリームをメインストリームに対して非同期に繰り返し再生するか否かを示すフラグとが記録され、フラグに基づきサブストリームの再生を制御できるようにしているため、メインストリームを再生中に、サブストリームをメインストリームに対して非同期に繰り返し再生することができる効果がある。

30

【 0 0 1 6 】

また、この発明は、ビデオデータとビデオデータに対応して再生されるオーディオデータと、ビデオデータとオーディオデータとが多重化されているか、それぞれ独立的なファイルとされているかを少なくとも示すフラグとが記録され、ビデオデータとオーディオデータを、フラグに基づき再生できるようにしているため、ビデオデータおよびオーディオデータの記録媒体からの読み出し方法をフラグに基づき制御することができる効果がある。

40

【図面の簡単な説明】

【 0 0 1 7 】

【図1】AVストリームファイルの再生順序指定の仕組みを模式的に示す略線図である。

【図2】クリップAV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UML図である。

【図3】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4】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5】ファイル「info.bdav」の一例の構造を表すシンタクスを示す略線図である。

50

【図 6】ブロックUIAppInfoBDVAV()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TableOfPlayLists()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「#####.rpls」および「#####.vpls」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UIAppInfoPlayList()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】ブロックPlayList()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PlayItem()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】ブリッジリップを説明するための略線図である。

【図 13】ブロックPlayListMark()の一例の構造を表すシンタクスを示す略線図である。

【図 14】ファイル「%%%%.clpi」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】この発明の実施の一形態で画像の表示系として用いられるプレーン構造の一例を示す略線図である。

【図 16】動画プレーン、字幕プレーンおよびグラフィクスプレーンの一例の解像度および表示可能色を示す略線図である。

【図 17】動画プレーン、字幕プレーンおよびグラフィクスプレーンを合成する一例の構成を示すブロック図である。

【図 18】パレットの入出力データの一例を示す略線図である。

【図 19】パレットに格納される一例のパレットテーブルを示す略線図である。

【図 20】グラフィクスプレーンに表示されるメニュー画面の一例を示す略線図である。

【図 21】独自に定義されたシナリオ記述言語を用いた場合のシナリオの一例の内部構造を示す略線図である。

【図 22】シナリオの構成の分類を説明するための略線図である。

【図 23】シナリオの構成の分類を説明するための略線図である。

【図 24】タイトルおよびチャプタについて説明するための略線図である。

【図 25】BD仮想プレーヤモデルについて説明するための略線図である。

【図 26】シナリオで記述されるコマンドによるBD仮想プレーヤ30の動作を概略的に示すフローチャートである。

【図 27】プレイリストの再生動作を説明するためのフローチャートである。

【図 28】シナリオで用いられるコマンドの一例を示す略線図である。

【図 29】この実施の一形態による一例のファイルの管理構造を示す略線図である。

【図 30】ファイル「scenario.hdmv」の一例の構造を表すシンタクスを示す略線図である。

【図 31】ブロックAutoplay()の一例のデータ構造を表すシンタクスを示す略線図である。

【図 32】ブロックScenario()の一例の構造を表すシンタクスを示す略線図である。

【図 33】ファイル「entrylist.data」の一例のデータ構造を表したシンタクスを示す略線図である。

【図 34】ブロックAppInfo()の一例の構造を表すシンタクスを示す略線図である。

【図 35】ブロックScenarioEntry()の一例の構造を表すシンタクスを示す略線図である。

【図 36】ファイル「xxxx.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 37】ブロックPLControlInfo()の一例の構造を表すシンタクスを示す略線図である。

【図 38】フィールドPL_playback_typeについて説明するための図である。

【図 39】フィールドPL_random_access_modeについて説明するための図である。

【図 40】ブロックPlayList()の一例の構造を表すシンタクスを示す略線図である。

【図 41】ブロックPlayItem()の一例の構造を表すシンタクスを示す略線図である。

【図 42】フィールドPI_random_access_modeについて説明するための図である。

10

20

30

40

50

- 【図 4 3】フィールド `still_mode` について説明するための図である。
- 【図 4 4】フィールド `is_seamless_angle_change` について説明するための図である。
- 【図 4 5】ブロック `SubPlayItem()` の一例の構造を表すシンタクスを示す略線図である。
- 【図 4 6】フィールド `is_repeat_flag` について説明するための図である。
- 【図 4 7】サブプレイアイテムのメインのプレイアイテムに対する同期再生について説明するための図である。
- 【図 4 8】ファイル「`zzzzz.clpi`」の一例の構造を表すシンタクスを示す略線図である。
- 【図 4 9】ブロック `ClipInfo()` の一例の構造を表すシンタクスを示す略線図である。
- 【図 5 0】フィールド `application_type` について説明するための図である。
- 【図 5 1】ブロック `SequenceInfo()` の一例の構造を表すシンタクスを示す略線図である。 10
- 【図 5 2】ブロック `ProgramInfo()` の一例の構造を表すシンタクスを示す略線図である。
- 【図 5 3】ブロック `StreamCodingInfo()` の一例の構造を表すシンタクスを示す略線図である。
- 【図 5 4】ブロック `CPI()` の一例の構造を表すシンタクスを示す略線図である。
- 【図 5 5】フィールド `CPI_type` について説明するための図である。
- 【図 5 6】ブロック `EP_map_for_HDMV()` の一例のデータ構造を表すシンタクスを示す略線図である。
- 【図 5 7】エクステンツについて説明するための図である。
- 【図 5 8】クリップ AV ストリームがディスク上に断片化して記録されている状態の例を示す略線図である。 20
- 【図 5 9】マルチアングル機能について説明するための図である。
- 【図 6 0】一つのインタリーブユニット中に複数のアングル切り替え可能点を設けることを説明するための図である。
- 【図 6 1】フラグ `is_angle_change_point` について説明するための図である。
- 【図 6 2】ノンシームレスのマルチアングルブロックの例を示す略線図である。
- 【図 6 3】ノンシームレスのマルチアングルブロックでアングル切り替えを行ったときに不連続点が生じてしまったことを説明するための図である。
- 【図 6 4】ノンシームレスのマルチアングルブロックの第 1 の再生方法について説明するための図である。
- 【図 6 5】ノンシームレスのマルチアングルブロックの第 2 の再生方法について説明するための図である。 30
- 【図 6 6】この発明の実施の一形態に適用できるプレーヤデコーダ 100 の一例の構成を示す機能ブロック図である。
- 【発明を実施するための形態】
- 【0018】
- 以下、この発明の実施の一形態について説明する。この発明の実施の一形態では、記録再生規格である `Blu-ray Disc` 規格 (Blu-ray Disc Rewritable Format Ver1.0) を元に、再生専用ディスクで必要となるインタラクティブ機能や、マルチアングル機能などの機能を実現できるようにした。
- 【0019】 40
- 先ず、理解を容易とするために、`Blu-ray Disc` に関し、"Blu-ray Disc Rewritable Format Ver1.0 part3 Audio Visual Specifications" で規定されている、コンテンツすなわちディスクに記録された AV (Audio/Video) データの管理構造について説明する。以下では、この管理構造を `BDAV` フォーマットと称する。
- 【0020】
- 例えば `MPEG` (Moving Pictures Experts Group) ビデオや `MPEG` オーディオなどの符号化方式で符号化され、`MPEG 2` システム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、`Blu-ray Disc` に関する規格の一つである "Blu-ray Disc Rewritable Format Ver1.0 part2" で定義されたファイルシステムにより、ファイルとしてディスクに記 50

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。

【0021】

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を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。Blu-ray Discに関する規格の一つである"Blu-ray Disc Rewritable Format Ver1.0 part3"で、このデータベースが規定される。

10

【0022】

図1は、AVストリームファイルの一部または全部の範囲を指定して、必要な部分だけを並べて再生する再生順序指定の仕組みを模式的に示す。図1において、プレイリスト(Playlist)は、AVストリームファイルの一部または全部の範囲を指定して、必要な部分だけを再生する指示を与える。ユーザがコンテンツの再生を行う際には、このプレイリストの単位で選択する。つまり、プレイリストは、ユーザからみてひとまとまり、すなわち、再生が連続に行われることをユーザが暗黙に期待する、映像・音声の単位である。

【0023】

最も簡単なプレイリストの構成は、記録が開始されてから記録が終了されるまでの一貫がりのAVストリームファイルからなり、編集をしなければ、これが一つのプレイリストになる。

20

【0024】

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と再生終了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。プレイアイテムは、一つの再生単位であると考えられる。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事は、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を再生するという事になる。

【0025】

クリップAVストリームは、上述したように、ビデオデータやオーディオデータがMP EG 2 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。

30

【0026】

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。

【0027】

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、クリップ中のエントリーポイントは、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。

40

【0028】

1のディスク中に記録された全てのプレイリストおよびクリップは、ボリュームインフォメーション(Volume Information)で管理される。

【0029】

図2は、上述のようなクリップAVストリーム、クリップ情報(Stream Attributes)、

50

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UML(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/または終了点異なる複数のプレイアイテムに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AV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AV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AV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。

【0030】

また、図3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。図3の例では、クリップ1が2つのプレイリスト2および3から参照されている。図3において、クリップ1は、横方向が時間軸を表す。プレイリスト2により、コマースル区間b-cとシーンeを含むクリップ1の区間a~fが参照される。一方、プレイリスト3により、シーンeを含むクリップ1の区間d-gが参照される。プレイリスト2を指定することで、クリップ1の区間a~fを再生することができ、プレイリスト3を指定することで、クリップ1の区間d-gを再生することができる。

【0031】

次に、"Blu-ray Disc Rewritable Format Ver1.0 part3"で規定された、記録媒体に記録されるファイルの管理構造について、図4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、先ず、1つのディレクトリ(図4の例ではルートディレクトリ)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。

【0032】

ルートディレクトリの下に、ディレクトリBD AVが置かれる。ディレクトリBD AVは、図4のように、ディレクトリBD AV、BD AV 1、BD AV 2、・・・、BD AV nの如く、ルートディレクトリの下に複数を置くことができる。以下では、複数のディレクトリBD AV、BD AV 1、BD AV 2、・・・、BD AV nをディレクトリBD AVで代表させて説明する。

【0033】

ディレクトリBD AVの下には、以下の6種類のファイルが置かれる。

- (1) info.bdav
- (2) menu.tidx、mark.tidx
- (3) menu.tdt1、menu.tdt2、mark.tdt1、mark.tdt2
- (4) #####.rpls、#####.vpls
- (5) %%%%.clpi
- (6) *****.m2ts

【0034】

なお、(4)のファイル「#####.rpls」および「#####.vpls」において、「#####」は、任意の番号を示す。同様に、(5)のファイル「%%%.clpi」において、「%%%.clpi」は任意の番号を示す。また、(6)のファイル「*****.m2ts」において、「*****」は、ファイル「*****.m2ts」がファイル「%%%.clpi」と一対一に対応するような番号である。例えば、番号「*****」は、番号「%%%.clpi」と同一とすることができる。

【0035】

(1)のファイル「info.bdav」は、ディレクトリBD AV全体の情報が格納されるファイルである。(2)のファイル「menu.tidx」および「mark.tidx」は、サムネイル画像の情報が格納されるファイルである。(3)のファイル「menu.tdt1」、「menu.tdt2」、「mark.tdt1」および「mark.tdt2」は、サムネイル画像が格納されるファイルである。各ファイルの拡張子「tdt1」および「tdt2」は、当該ファイルに格納されているサムネイル画像データが暗号化されているか否かを示す。

10

20

30

40

50

【 0 0 3 6 】

(4) のファイル「#####.rpls」、「#####.vpls」は、プレイリストの情報が格納されるファイルである。ファイル「#####.rpls」、「#####.vpls」は、ディレクトリ B D A V の下に設けられたディレクトリ P L A Y L I S T のさらに下に置かれている。

【 0 0 3 7 】

(5) のファイル「%%%%.clpi」は、クリップ情報が格納されるファイルである。ファイル「%%%%.clpi」は、ディレクトリ B D A V の下に設けられたディレクトリ C L I P I N F のさらに下に置かれている。(6) のファイル「*****.m2ts」は、クリップ A V ストリームが格納される、クリップ A V ストリームファイルである。このクリップ A V ストリームファイルは、ファイル名の番号「*****」により、1つのクリップ情報ファイル「%%%%.clpi」に対応付けられる。ファイル「*****.m2ts」は、ディレクトリ B D A V の下に設けられたディレクトリ S T R E A M のさらに下に置かれる。

【 0 0 3 8 】

各ファイルについて、より詳細に説明する。(1) のファイル「info.bdav」は、ディレクトリ B D A V の下に唯一、設けられる。図 5 は、ファイル「info.bdav」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 3 9 】

図 5 において、ファイル「info.bdav」の内部は、機能別の情報毎にブロックが構成される。フィールド type_indicator は、文字列「BD AV」が格納され、このファイルがファイル「info.bdav」であることが示される。フィールド version_number は、このファイル「info.bdav」のバージョンが示される。ブロック UIAppInfoBD AV() は、ディレクトリ B D A V の下に関する情報が格納される。ブロック TableOfPlayLists() は、プレイリストの並びに関する情報が格納される。ブロック MakersPrivateData() は、記録再生装置のメーカー固有の情報が格納される。

【 0 0 4 0 】

ファイル「info.bdav」の先頭には、各ブロックの先頭を表すアドレスが記述される。例えば、フィールド TableOfPlayLists_Start_address は、ブロック「TableOfPlayLists()」の開始する位置がファイル内の相対バイト数で示される。

【 0 0 4 1 】

図 6 は、ブロック UIAppInfoBD AV() の一例の構造を表すシンタクスを示す。フィールド length は、フィールド length 直後のフィールドからこのブロック UIAppInfoBD AV() の最後までまでの長さがバイトで示される。フィールド BD AV_character_set は、このブロック UIAppInfoBD AV() 内のフィールド BD AV_name に記録されている文字列の文字セットが示される。文字セットとしては、A S C I I、U n i c o d e などが選択できる。

【 0 0 4 2 】

フラグ BD AV_protect_flag は、ディレクトリ B D A V の下に置かれるコンテンツを、ユーザに制限無しに見せて良いか否かを示す。例えば、このフラグが値「1」にセットされており、且つ、ユーザが正しく暗証番号 PIN(Personal Identification Number) を入力できた場合に、ディレクトリ B D A V の下に置かれたコンテンツをユーザに見せることができるものとする。一方、フラグ BD AV_protect_flag が値「0」にセットされている場合には、ユーザによる暗証番号 PIN の入力無くても、ディレクトリ B D A V の下に置かれたコンテンツをユーザに見せることができる。

【 0 0 4 3 】

暗証番号 PIN は、フィールド PIN に記録される。暗証番号 PIN は、例えば 4 桁の 0 ~ 9 の数字からなり、上述のように再生制御を有効にした場合に必要となる暗証番号を表す。暗証番号 PIN のそれぞれの数字は、例えば I S O (International Organization for Standardization) / I E C (International Electrotechnical Commission) 6 4 6 の規定に従って符号化される。

10

20

30

40

50

【 0 0 4 4 】

このブロックUIAppInfoBDV()に示される以上の情報により、ディレクトリBDVに対する再生制限が規定される。なお、詳細は後述するが、個々のプレイリストに対する再生制限については、ファイル「#####.rpls」、「#####.vpls」内に示されるブロックUIAppInfoPlayList()の中で定義されるフラグplayback_control_flagによって規定される。

【 0 0 4 5 】

この例では、ディレクトリBDVの下のコンテンツの再生を再開する際に、優先して再生すべきプレイリストを指定する、レジューム機能を用いることができる。このレジューム機能は、前回視聴を中断した箇所から再生を再開したい場合などに使うことを想定している。

10

【 0 0 4 6 】

図6において、フラグresume_valid_flagは、このレジューム機能の有効/無効を示す。例えば、このフラグが値「0」にセットされている場合には、レジューム機能は無効とされ、このフラグが値「1」にセットされている場合には、レジューム機能が有効とされ、その際、フィールドresume_PlayList_file_nameで指定されるプレイリストを、優先して再生すべきプレイリストとして扱う。

【 0 0 4 7 】

フィールドref_to_menu_thumbnail_indexは、このディレクトリBDVを代表するサムネイル画像が存在する場合、そのサムネイル画像を特定するサムネイル番号が格納される領域である。Blu-ray Discの規格では、ディレクトリBDVを代表する静止画を特にメニューサムネイルと呼ぶ。フィールドref_to_menu_thumbnail_indexで指定されるインデックスthumbnail_indexを持つサムネイル画像がこのディレクトリBDVのメニューサムネイルとなる。

20

【 0 0 4 8 】

フィールドBDV_name_lengthは、フィールドBDV_nameに示されるディレクトリBDV名のパイト長を示す。このフィールドBDV_nameにおいて、左から、フィールドBDV_name_lengthで示されるだけのパイト数が有効な文字列であり、それがこのディレクトリBDVの名前を示す。なお、フィールドBDV_nameの中で、フィールドBDV_name_lengthで示された有効な文字列の後のパイト列には、どのような値が入っていても良い。

【 0 0 4 9 】

図7は、ブロックTableOfPlayLists()の一例の構造を表すシンタクスを示す。フィールドnumber_of_PlayListsは、このディレクトリBDVの下に置かれたプレイリストの数が示される。このプレイリスト数をループ変数とする、次に続く「for」文のループにおいてフィールドPlayList_file_nameにより示されるプレイリストの並びが、プレイリスト一覧表示画面などに表示される際のプレイリストの順序を示す。プレイリストは、フィールドPlayList_file_nameにより、「#####.rpls」や「#####.vpls」といったファイル名で指定される。

30

【 0 0 5 0 】

ファイル「#####.rpls」および「#####.vpls」は、上述したように、ディレクトリPLAYLISTの下に置かれる。これらのファイルは、1つのプレイリストと一対一に対応している。

40

【 0 0 5 1 】

図8は、ファイル「#####.rpls」および「#####.vpls」の一例の構造を表すシンタクスを示す。図8において、ファイル「#####.rpls」および「#####.vpls」の内部は、機能別の情報毎にブロックが構成される。フィールドtype_indicatorにこのファイルを示す文字列が格納され、フィールドversion_numberにこのファイルのバージョンが示される。

【 0 0 5 2 】

ブロックUIAppInfoPlayList()は、このプレイリストに関する属性情報が格納される。ブロックPlayList()は、このプレイリストを構成するプレイアイテムに関する情報が格納される。ブロックPlayListMark()は、このプレイリストに付されるマークの情報が格納さ

50

れる。ブロックMakersPrivateData()は、このプレイリストファイルを記録した装置の、メーカ固有の情報が格納される。ファイル「#####.rpls」および「#####.vpls」の先頭に設けられるフィールドPlaylist_start_address、PlaylistMark_start_addressおよびMakersPrivateData_start_addressには、それぞれ対応するブロックの先頭アドレスが例えば32ビットのアドレス情報として示される。

【0053】

各ブロックの先頭アドレスがファイル「#####.rpls」および「#####.vpls」の先頭に示されるために、ブロックの前および/または後ろに、任意の長さのデータpadding_wordを挿入することができる。但し、ファイル「#####.rpls」および「#####.vpls」の最初のブロックであるブロックUIAppInfoPlaylist()の開始位置は、ファイルの先頭から320バ

10

【0054】

図9は、ブロックUIAppInfoPlaylist()の一例の構造を表すシンタクスを示す。ブロックUIAppInfoPlaylist()は、プレイリストの再生において直接的に必要とされない、プレイリストに関する各種の属性情報が格納される。フィールドPlaylist_character_setは、プレイリストに関する文字列情報の文字セットが指定される。

【0055】

フラグplayback_control_flagは、情報表示やプレイリストの再生を暗証番号PINに基づき制限するか否かが指定される。例えばフラグplayback_control_flagが値「1」であれば、ユーザにより正しい暗証番号PINが入力されなければ、プレイリストのサムネイル画

20

【0056】

フィールドtime_zoneは、このプレイリストが記録された地点のタイムゾーンを示す。フィールドrecord_time_and_dateは、プレイリストの記録日時を示す。フィールドPlaylist_durationは、プレイリストの再生時間を示す。

【0057】

フィールドmaker_IDおよびmaker_model_codeは、このプレイリストを最終更新した記録機のメーカおよび記録機をそれぞれ特定する情報が例えば番号で示される。フィールドchannel_numberは、記録したクリップAVストリームのチャンネル番号が示され、フィールドchannel_nameは、当該チャンネル名が示される。フィールドchannel_nameに示されるチャンネル名の長さがフィールドchannel_name_lengthに示される。フィールドchannel_nameの領域のうち、フィールドchannel_name_lengthで示される長さの文字列が有効となる。フィールドPlaylist_nameは、フィールドPlaylist_name_lengthに示される値を有効長としてプレイリスト名が示される。フィールドPlaylist_detailは、フィールドPlaylist_detail_lengthに示される値を有効長としてプレイリストの詳細情報が示される。

30

【0058】

図10は、ブロックPlaylist()の一例の構造を表すシンタクスを示す。フィールドlengthは、フィールドlengthの直後のフィールドからこのブロックPlaylist()の終端までのバイト長を示す。フィールドPL_CPI_typeは、このプレイリストが持つCPI(Characteristic Point Information: 特徴点情報)の種類を示す。フィールドnumber_of_PlayItemsは、このプレイリストを構成するプレイアイテムの数を示す。フィールドnumber_of_SubPlayItemsは、このプレイリストに付けられているアフレコオーディオ用のプレイアイテム(サブプレイアイテム)の数を示す。詳細は省略するが、プレイリストは、所定の条件を満たす場合にだけ、サブプレイアイテムを持つことができる。

40

【0059】

ブロックPlayItem()は、プレイアイテムの情報が格納される。ブロックSubPlayItem()

50

は、サブプレイアイテムの情報が格納される。

【 0 0 6 0 】

図 1 1 は、ブロックPlayItem()の一例の構造を表すシンタクスを示す。フィールドClip_Information_file_nameは、このプレイアイテムが参照しているクリップと一対一に対応するクリップ情報ファイル(拡張子がclpiであるファイル)のファイル名が文字列で示される。クリップ情報ファイルは、拡張子が「clpi」とされているファイルである。

【 0 0 6 1 】

フィールドClip_codec_identifierは、このプレイアイテムにより参照されるクリップの符号化方式を示す。この例では、フィールドClip_codec_Identifierは、値「M2TS」に固定的とされる。フィールドconnection_conditionは、このプレイアイテムが次のプレイアイテムとの間でどのような接続がされているかを示す情報である。すなわち、フィールドconnection_conditionにより、プレイアイテムとプレイアイテムとの間が継ぎ目なくシームレスに再生できるか否かが示される。

【 0 0 6 2 】

フィールドref_to_STC_idは、このプレイアイテムにより参照されるクリップ内のシーケンスSTC_sequenceを指定する。シーケンスSTC_sequence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Bluray Disc規格独自の構造である。シーケンスSTC_sequence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STC_sequence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STC_sequenceに存在していなければならない。フィールドref_to_STC_idでは、番号STC_idによりシーケンスSTC_sequenceが指定される。

【 0 0 6 3 】

フィールドIN_timeおよびOUT_Timeは、このプレイアイテムにおける開始点および終了点の、シーケンスSTC_sequence 上でのタイムスタンプpts(presentation_time_stamp)をそれぞれ示す。

【 0 0 6 4 】

ブロックBridgeSequenceInfo()は、ブリッジクリップ(Bridge-Clip)に関する情報が格納される。ブリッジクリップは、図 1 2 に一例が示されるように、プレイアイテム間を継ぎ目なく、シームレスに再生する機能を実現する際に作成されるビットストリームである。前のプレイアイテムと現在のプレイアイテムとの継ぎ目の部分で、本来再生すべきビットストリームとは異なるビットストリーム、すなわちブリッジクリップを代わりに再生することで、2つのプレイアイテムをシームレスに再生することができる。このブリッジクリップによる機能は、この発明と関連性が低いので、詳細は省略する。

【 0 0 6 5 】

図 1 3 は、ブロックPlayListMark()の一例の構造を表すシンタクスを示す。ブロックPlayListMark()は、マークの情報が格納されるデータ構造である。マークは、プレイリスト上の時刻を保持するための構造であって、マークを用いることにより、プレイリスト中への頭出し点の設定、プレイリストを細分化するチャプタの構造を設ける、などの機能が実現される。さらに、後述するグラフィクスプレーン上に描画された画像の表示開始/停止のタイミングを、マークを用いて指定することができる。

【 0 0 6 6 】

フィールドlengthは、フィールドlengthの直後のフィールドからブロックPlayListMark()の終端までのバイト長を示す。フィールドnumber_of_PlayList_marksは、プレイリスト中のマークの数を表す。「for」文での1回のループが1つのマークの情報を示す。フラグmark_invalid_flagは、このマークが有効であるか否かを示す。例えば、フラグmark_invalid_flagが値「0」でそのマークが有効であることが示され、値「1」であれば、そのマークの情報がデータベース上に存在するが、当該マークがユーザには見えない無効マークとされていることを示す。

10

20

30

40

50

【 0 0 6 7 】

フィールドmark_typeは、このマークの種類を表す。マークの種類としては、プレイリストのサムネイル画像（代表画）とする映像の位置を示すマーク、どこまで再生したかを示すレジュームマーク、頭出し点を示すチャプタマーク、ある区間の再生を飛ばして先に進むことを示すスキップマーク、グラフィクス画像の読み込み開始タイミングを示すマーク、グラフィクス画像の表示開始のタイミングを示すマーク、グラフィクス画像の表示停止のタイミングを示すマークなどがある。

【 0 0 6 8 】

フィールドmark_name_lengthは、後述するフィールドmark_nameのデータ長を示す。フィールドmaker_IDは、このマークを作成した記録機の製造者を示す。これは、製造者独自のマークを識別する際に用いられる。フィールドref_to_PlayItem_idは、このマークにより指示される時刻がどのプレイアイテム上に存在するかが示される。フィールドmark_time_stampは、このマークにより指示される時刻が示される。

10

【 0 0 6 9 】

フィールドentry_ES_PIDは、このマークがどのエレメンタリストリームすなわち映像データおよび/または音声データが符号化されたストリームに対して付与されたものかが示される。フィールドref_to_menu_thumbnail_indexおよびフィールドref_to_mark_thumbnail_indexは、マークを視覚的に表すサムネイル画像を示す。サムネイル画像としては、例えばそのマークにより指定される時刻の映像を静止画として抜き出した画像が考えられる。

20

【 0 0 7 0 】

フィールドdurationは、マークが時間軸上で長さを持つ場合に使われる。例えばスキップマークであれば、どのくらいの時間ジャンプをするかがフィールドdurationにより指定される。

【 0 0 7 1 】

フィールドmakers_informationは、製造者独自の情報を記録するための領域である。フィールドmark_nameは、マークに名前をつけた際、その名前を格納するための領域であり、サイズが上述のフィールドmark_name_lengthで指定される。

【 0 0 7 2 】

図 1 4 は、ファイル「%%%%.clpi」の一例の構造を表すシンタクスを示す。上述したように、ファイル「%%%%.clpi」は、ディレクトリCLIPINFの下に置かれ、各AVストリームファイル（ファイル「****.m2ts」）毎に作られる。ファイル「%%%%.clpi」の内部は、機能別の情報毎にブロックが構成される。フィールドtype_indicatorにこのファイルを示す文字列が格納され、フィールドversion_numberにこのファイルのバージョンが示される。

30

【 0 0 7 3 】

ブロックClipInfo()は、クリップに関する情報が格納される。ブロックSequenceInfo()は、MPEG2システムにおけるトランスポートストリームの時刻基準を表すPCRの不連続点に関する情報が格納される。ブロックProgramInfo()は、MPEG2システムのプログラムに関する情報が格納される。ブロックCPI()は、ランダムアクセス開始可能点などの、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。ブロックClipMark()は、クリップに付された頭出しのためのインデックス点やコマシャルの開始および/または終了点などのマーク情報が格納される。ブロックMakersPrivateData()は、記録機の製造者独自の情報が格納される。

40

【 0 0 7 4 】

また、これら各ブロックのファイル「%%%%.clpi」中での先頭を表すアドレス情報が、当該ファイルの先頭部分にフィールドSequenceInfo_start_address、ProgramInfo_start_address、CPI_start_address、ClipMark_start_addressおよびMakersPrivateData_start_addressとして示される。なお、このクリップファイル「%%%%.clpi」自体は、この発明と関連性が低いので、詳細な説明を省略する。

50

【0075】

BDAVフォーマットは、以上に概略的に説明したようなデータ構造を持つことにより、クリップAVストリーム中の再生したい部分を開始および終了点の組で指定したプレイアイテムの並びでプレイリストを構築し、ユーザが認識するひとまとまりの再生単位を管理することが可能となる。

【0076】

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、上述したBDAVフォーマットを拡張し、再生専用ディスクに適したフォーマットを構築する。まずディスクの内容を提示するメニュー画面を実現するためのプレーン構成を説明する、次に、コンテンツ制作者側がプレイリストの再生順序を指定できるようにするためのシナリオ構造を追加する。さらに、そのシナリオ構造において、再生専用ディスクに特徴的なスチル（一時停止）、ランダム・シャッフル再生、マルチアングルなどの機能を実現するために必要なデータと、その格納方法について説明する。

10

【0077】

この発明の実施の一形態では、画像の表示系について、図15に一例が示されるようなプレーン構成を取る。動画プレーン10は、最も後ろ側（ボトム）に表示され、プレイリストで指定された画像（主に動画データ）が扱われる。字幕プレーン11は、動画プレーン10の上に表示され、動画再生中に表示される字幕データが扱われる。グラフィクスプレーン12は、最も前面に表示され、メニュー画面を表示するための文字データやボタンを表すビットマップデータなどのグラフィクスデータが扱われる。1つの表示画面は、これら3つのプレーンが合成されて表示される。

20

【0078】

ここで、従来のDVDビデオと異なる特徴的な点は、従来の技術で説明した字幕やメニュー画面、ボタンなどが表示されるサブピクチャを、字幕プレーン11およびグラフィクスプレーン12とに分離し、字幕とボタンとをそれぞれ独立的に制御できるようにしたことである。上述したように、従来のDVDビデオでは、メニュー画面やボタンなどのグラフィクス表示と、字幕表示とを、同じ仕組みで制御しており、これらを同一プレーン上に表示していた。そして、同時に表示できるビットマップ画像は、1枚に限られていた。そのため、従来のDVDビデオでは、複数のビットマップ画像を同時に表示することができなかった。この発明では、字幕を表示する字幕プレーン11と、グラフィクス表示を行うグラフィクスプレーン12とをそれぞれ独立的に設けることで、この従来のDVDビデオの持つ問題点を解消している。

30

【0079】

なお、画像の表示系に関しては、字幕プレーン11およびグラフィクスプレーン12が、従来の"Blu-ray Disc Rewritable Format Ver1.0 part3"に対する拡張部分であると考えることができる。

【0080】

動画プレーン10、字幕プレーン11およびグラフィクスプレーン12は、それぞれ独立して表示が可能とされ、例えば、図16に一例が示されるような解像度および表示可能色を有する。動画プレーン10は、解像度が1920画素×1080ラインで1画素当たり

40

【0081】

グラフィクスプレーン12は、解像度が1920画素×1080ラインで各画素のサンプリング深さが8ビットとされ、カラーシステムは、256色のパレットを用いた8ビットカラーマップアドレスとされる。字幕プレーン11は、1920画素×1080ラインで各画素のサンプリング深さが8ビットとされ、カラーシステムは、256色のパレット

50

を用いた 8 ビットカラーマップアドレスとされる。

【 0 0 8 2 】

グラフィクスプレーン 1 2 および字幕プレーン 1 1 は、2 5 6 段階のアルファブレンディングが可能とされており、他のプレーンとの合成の際に、不透明度を 2 5 6 段階で設定することが可能とされている。不透明度の設定は、画素毎に行うことができる。以下では、不透明度 が (0 1) の範囲で表され、不透明度 = 0 で完全に透明、不透明度 = 1 で完全に不透明であるものとする。

【 0 0 8 3 】

字幕プレーン 1 1 では、P N G (Portable Network Graphics) 形式の画像データが扱われる。また、グラフィクスプレーン 1 2 でも、P N G 形式の画像データを扱うことができる。P N G 形式は、1 画素のサンプリング深さが 1 ビット ~ 1 6 ビットとされ、サンプリング深さが 8 ビットまたは 1 6 ビットの場合に、アルファチャンネル、すなわち、それぞれの画素成分の不透明度情報 (アルファデータと称する) を付加することができる。サンプリング深さが 8 ビットの場合には、2 5 6 段階で不透明度を指定することができる。このアルファチャンネルによる不透明度情報を用いてアルファブレンディングが行われる。また、2 5 6 色までのパレットイメージを用いることができ、予め用意されたパレットの何番目の要素 (インデックス) であるかがインデックス番号により表現される。

【 0 0 8 4 】

なお、字幕プレーン 1 1 およびグラフィクスプレーン 1 2 で扱われる画像データは、P N G 形式に限定されない。J P E G 方式など他の圧縮符号化方式で圧縮符号化された画像データや、ランレングス圧縮された画像データ、圧縮符号化がなされていないビットマップデータなどを扱うようにしてもよい。

【 0 0 8 5 】

図 1 7 は、上述の図 1 5 および図 1 6 に従い 3 つのプレーンを合成する一例の構成を示す。動画プレーン 1 0 の動画データが 4 2 2 / 4 4 4 変換回路 2 0 に供給される。動画データは、4 2 2 / 4 4 4 変換回路 2 0 でカラーシステムが Y C b C r (4 : 2 : 2) から Y C b C r (4 : 4 : 4) に変換され、乗算器 2 1 に入力される。

【 0 0 8 6 】

字幕プレーン 1 1 の画像データがパレット 2 2 に入力され、R G B (4 : 4 : 4) の画像データとして出力される。アルファブレンディングによる不透明度が指定されている場合には、指定された不透明度 $1 (0 < = \alpha < = 1)$ がパレット 2 2 から出力される。

【 0 0 8 7 】

図 1 8 は、パレット 2 2 の入出力データの一例を示す。パレット 2 2 は、例えば P N G 形式のファイルに対応したパレット情報がテーブルとして格納される。パレット 2 2 は、入力された 8 ビットの画素データをアドレスとして、インデックス番号が参照される。このインデックス番号に基づき、それぞれ 8 ビットのデータからなる R G B (4 : 4 : 4) のデータが出力される。それと共に、パレット 2 2 では、不透明度を表すアルファチャンネルのデータが取り出される。

【 0 0 8 8 】

図 1 9 は、パレット 2 2 に格納される一例のパレットテーブルを示す。2 5 6 個のカラーインデックス値 $\{ 0 \times 0 0 \} \sim \{ 0 \times F F \}$ ($\{ 0 \times \}$ は 1 6 進表記であることを示す) のそれぞれに対して、各々 8 ビットで表現される三原色の値 R、G および B と、不透明度 α とが割り当てられる。パレット 2 2 は、入力された P N G 形式の画像データに基づきパレットテーブルが参照され、画像データにより指定されたインデックス値に対応する、それぞれ 8 ビットのデータからなる R、G および B 各色のデータ (R G B データ) と、不透明度 α とを画素毎に出力する。

【 0 0 8 9 】

パレット 2 2 から出力された R G B データは、R G B / Y C b C r 変換回路 2 9 に供給され、各データ長が 8 ビットの輝度信号 Y と色信号 C b、C r のデータに変換される (以下、まとめて Y C b C r データと称する)。これは、以降のプレーン間合成を共通のデー

10

20

30

40

50

タ形式で行う必要があるため、動画データのデータ形式である Y C b C r データに統一している。

【 0 0 9 0 】

R G B / Y C b C r 変換回路 2 9 から出力された Y C b C r データおよび不透明度データ 1 とがそれぞれ乗算器 2 3 に入力される。乗算器 2 3 では、入力された Y C b C r データに不透明度データ 1 が乗ぜられる。乗算結果は、加算器 2 4 の一方の入力端に入力される。なお、乗算器 2 3 では、Y C b C r データにおける輝度信号 Y、色差信号 C b、C r のそれぞれについて、不透明度データ 1 との乗算が行われる。また、不透明度データ 1 の補数 (1 - 1) が乗算器 2 1 に供給される。

【 0 0 9 1 】

乗算器 2 1 では、4 2 2 / 4 4 4 変換回路 2 0 から入力された動画データに不透明度データ 1 の補数 (1 - 1) が乗ぜられる。乗算結果は、加算器 2 4 の他方の入力端に入力される。加算器 2 4 において、乗算器 2 1 および 2 3 の乗算結果が加算される。これにより、動画プレーン 1 0 と字幕プレーン 1 1 とが合成される。加算器 2 4 の加算結果が乗算器 2 5 に入力される。

【 0 0 9 2 】

グラフィクスプレーン 1 2 の画像データも字幕プレーン 1 1 と同様に、パレットテーブル 2 6 により R G B (4 : 4 : 4) のデータが出力され、R G B / Y C b C r 変換回路 2 6 に入力される。グラフィクスプレーン 1 2 の画像データのカラーシステムが R G B (4 : 4 : 4) である場合には、カラーシステムが Y C b C r (4 : 4 : 4) に変換されて R G B / Y C b C r 変換回路 2 7 から出力される。R G B / Y C b C r 変換回路 2 7 から出力された Y C b C r データが乗算器 2 8 に入力される。

【 0 0 9 3 】

グラフィクスプレーン 1 2 で用いられる画像データが P N G 形式である場合には、画像データ中に、画素毎に不透明度データ 2 (0 2 1) を設定することができる。不透明度データ 2 は、乗算器 2 8 に供給される。乗算器 2 8 では、R G B / Y C b C r 変換回路 2 7 から入力された Y C b C r データに対し、輝度信号 Y、色差信号 C b、C r のそれぞれについて、不透明度データ 2 との乗算が行われる。乗算器 2 8 による乗算結果が加算器 2 9 の一方の入力端に入力される。また、不透明度データ 2 の補数 (1 - 2) が乗算器 2 5 に供給される。

【 0 0 9 4 】

乗算器 2 5 では、加算器 2 4 の加算結果に対して不透明度データ 2 の補数 (1 - 2) が乗ぜられる。乗算器 2 5 の乗算結果は、加算器 2 7 の他方の入力端に入力され、上述した乗算器 2 8 による乗算結果と加算される。これにより、動画プレーン 1 0 と字幕プレーン 1 1 との合成結果に対して、さらに、グラフィクスプレーン 1 2 が合成される。

【 0 0 9 5 】

字幕プレーン 1 1 およびグラフィクスプレーン 1 2 において、例えば、表示すべき画像の無い領域の不透明度 = 0 と設定することで、そのプレーンの下に表示されるプレーンを透過表示させることができ、例えば動画プレーン 1 0 に表示されている動画データを、字幕プレーン 1 1 やグラフィクスプレーン 1 2 の背景として表示することができる。

【 0 0 9 6 】

なお、この図 1 7 に示される構成は、ハードウェアおよびソフトウェアの何れでも実現可能なものである。

【 0 0 9 7 】

以上のようなプレーンを設定することで、再生専用規格に必要なメニュー画面とボタンの表示を可能とする。メニュー画面上のボタンが選択されると、そのボタンに対応したプレイリストが再生されることになる。このとき、プレイリスト間がどのように結び付けられているかという情報がディスク上に記録されていることが必要となる。次に、このプレイリスト間の結びつき (リンク) を定義するシナリオについて説明する。

【 0 0 9 8 】

10

20

30

40

50

グラフィクスプレーン 12 には、ユーザに対して操作を促す画面、例えばメニュー画面を表示させることができる。図 20 は、グラフィクスプレーン 12 に表示されるメニュー画面 60 の一例を示す。メニュー画面 60 は、文字や画像を特定の位置に表示させ、ユーザが選択や実行を指示することで新たな動作が開始される「リンク」や「ボタン」を配置することができる。

【0099】

「リンク」は、例えば、文字列や画像データに対して所定のファイルへのアクセス方法を示す記述を設定し、ユーザがポインティングデバイスなどを用いて画面に表示された当該文字列や画像データ上を指定することで、当該文字列や画像データに設定されたアクセス方法に従って当該ファイルにアクセスできるようにしたものである。「ボタン」は、「リンク」において、例えば通常の状態、選択された状態および押された状態の3種類の画像データを用意し、ユーザにより当該ボタン画像が指定された際に、3種類の画像データの表示を切り換えて、ユーザが行った操作をユーザ自身に認識しやすくしたものである。

10

【0100】

なお、「リンク」や「ボタン」に対する指定動作は、例えば、マウスを用いて画面上のカーソル表示を移動させ、「リンク」による文字列や画像、「ボタン」による画像上で、マウスボタンを1または複数回押すクリック動作を行うことで、なされる。マウス以外のポインティングデバイスによっても、同様の動作を行うことができる。また、これに限らず、リモートコントロールコマンドやキーボードのキー操作により、「リンク」や「ボタン」の指定を行うこともできる。例えば、方向キーなど所定のキーで指定を行いたい「リンク」や「ボタン」を選択し、決定キーなどにより選択された「リンク」や「ボタン」を指定する。

20

【0101】

図 20 の例では、グラフィクスプレーン 12 上に表示されたメニュー画面 60 の上部に、画像データとしてタイトル 61 が表示される。タイトル 61 に続けて、リンクとしての選択項目 62 A、62 B、62 C および 62 D が表示される。例えばリモートコントロールコマンドのキー操作により、選択項目 62 A、62 B、62 C および 62 D のうち1つを選択し指定することで、指定された選択項目にリンクとして設定されたファイルがアクセスされる。

【0102】

メニュー画面 60 の下部に、字幕の表示の有無や、出力される音声を例えば英語および日本語から選択するためのボタン 64 および 65 が表示される。これらボタン 64 や 65 を上述のように操作することで、例えば、それぞれの設定を行う画面を表示するためのファイルがアクセスされ、所定の画面表示がなされる。

30

【0103】

また、メニュー画面 60 の左下部分には、項目の選択方法を示す文字列 63 が表示される。この文字列 63 も、グラフィクスプレーン 12 上の表示である。

【0104】

この、図 20 に一例が示されるような画面を表示するためには、画面の表示方法やリンク情報などを記述可能な、何らかの記述言語が必要とされる。この発明の実施の一形態では、この記述言語として、DVD ビデオのナビゲーションコマンドを基に変更を加えたものに、字幕、ボタン用の独自の表示制御コマンドを加えた命令群を使用することで、Blu-ray Disc によるメニュー画面表示が可能にしようとした。

40

【0105】

さて、上述したような Blu-ray Disc のメニュー画面 60 では、例えばプレイリストの一覧が画像データや文字列、ボタンなどで表示され、あるプレイリストを指定することで、指定されたプレイリストがディスクから読み出され再生されることが期待される。

【0106】

上述した図 20 の例では、メニュー画面 60 に対してプレイリストの一覧が表示される

50

。このメニュー画面60およびメニュー画面60上の選択項目を選択した際に再生される映像・音声は、実際には、複数のプレイリストが並んだ構造によって構成されている。メニューの一項目を構成する複数のプレイリスト内で、プレイリストの関連付けを行うことで、ストーリーを分岐させる仕組みを実現することができる。ストーリーを分岐できるようにすることで、ユーザの選択によってストーリーが変化していくマルチストーリー機能や、プレーヤの設定言語に従って自動的に適切な言語を再生する機能や、ユーザの年齢に応じてシーンを差し替えるパンタレル機能を実現することができる。

【0107】

このような機能は、記録済みのディスクにおいて特に有用であって、テレビジョン放送の記録/再生が主目的である現行のBlu-ray Disc規格では想定されていない。

10

【0108】

以下では、この複数のプレイリストが並べられた構造を、シナリオと称する。図21は、シナリオ70の一例の内部構造を示す。このシナリオ70は、複数のプレイリスト73A~73Mを含む。また、シナリオ70は、グラフィクスプレーン12を用いて分岐選択画面を表示する箇所が2箇所(画面80Aおよび80B)設けられている。例えば画面80Aは、分岐選択画面を表示するためのグラフィクスデータ74Aとプレイリスト73Cを有する。同様に、画面80Bは、分岐選択画面を表示するためのグラフィクスデータ74Bとプレイリスト73Jを有する。

【0109】

このように、シナリオは、プレイリストの並びを決めると共に、グラフィクスプレーン12に対する表示のタイミングを規定する。グラフィクスプレーン12への表示のタイミングは、グラフィクスプレーンに表示する画像に対して付加された表示制御命令によって規定することができる。

20

【0110】

図21の例では、例えば上述のメニュー画面60はシナリオ70における80Aに対応する。メニュー画面60における選択項目(例えば選択項目62A)は、グラフィクス72Aによって構成されている。すなわち、メニュー画面60において選択項目62Aが指定されると、選択項目に対応したプレイリスト73Dが再生されることになる。

【0111】

図21のシナリオ70では、まず、ディスクがプレーヤに挿入されると、プレイリスト73Aが再生される。プレイリスト73Aの再生が終了されると、続けてプレイリスト73Bが再生され、プレイリスト73Bの再生が終了されると、プレイリスト73Cが再生され、グラフィクスデータ72Aが読み出されて、ユーザにストーリーの分岐選択を促す画面80Aが表示される。

30

【0112】

画面80Aが表示された以降は、ユーザの選択に基づきストーリーが分岐される。この図20の例では、第1の選択により、画面80Aの後、プレイリスト73D、73E、73Fの順に再生され、全体の再生が終了される。プレイリスト73Fの再生が終了された後、メインのメニュー画面(例えば上述したメニュー画面60)に戻るようにしても良い。

40

【0113】

画面80Aにおける第2の選択では、画面80Aの後、プレイリスト73Gが再生される。プレイリスト73Gは、例えば所定のタイミングにマークが設定されている。プレイリスト73Gが再生されると、再生装置自身の設定やユーザによる他のシナリオや分岐選択画面などでの選択に基づき、プレイリスト73Gのマーク位置で分岐するか、プレイリスト73Gを最後まで再生するかが制御される。プレイリスト73Gが最後まで再生される場合には、プレイリスト73Gの再生終了後、プレイリスト73M、73Iと再生され、プレイリスト73Jがさらに再生される。

【0114】

50

一方、プレイリスト73Gにおいてマーク位置で分岐された場合には、次にプレイリスト73K、73Lと再生され、プレイリスト73Lの再生が終了されたら、次に、プレイリスト73Iに設定されたマーク位置から再生が継続される。

【0115】

プレイリスト73Jは、グラフィクスデータ72Bが読み出されて、ユーザにストーリーの分岐選択を促す画面80Bが表示される。画面80Bにおける第1の選択では、上述したプレイリスト73Fが再生される。一方、画面80Bの第2の選択では、プレイリスト73Kに設定されているマーク位置からの再生が行われる。

【0116】

シナリオの再生において、マークの検出、ユーザからの入力、プレーヤの動作変更の検出がされた時の動作は、プレイリスト毎にコマンド列(プログラム)が用意されており、そのプログラムをプレーヤが実行することにより行われる。

【0117】

例えば、シナリオ70内のプレイリスト73A~73Mの何れが再生中であっても、リモートコントロールコマンドに設けられたメニューボタンが押されたら、シナリオの一覧が表示されるメニュー画面60が表示される、すなわち、メニュー画面60を表示するためのプレイリストに再生処理が移行される動作を実現したい場合について考える。この場合、リモートコントロールコマンドのメニューボタンが押されたときに発生するイベント(メニューボタン押下イベント)に対応するイベントハンドラとして、メニュー画面60を表示させるプレイリストに処理が移行される命令を、シナリオデータの一つとして記述しておく。

【0118】

シナリオは、ディレクトリに一つ定義され、1または複数のプレイリストから構成される。ここでディレクトリとは、例えば上述した記録再生規格のBD-AVディレクトリや、この発明の実施の一形態において再生専用のデータ領域として想定しているHDMVディレクトリ(詳細は後述する)である。

【0119】

ここで、シナリオの構成の分類について図22および図23を用いて説明する。シナリオの構成は、プレイリスト同士の接続を基に考えると、図22に示されるように、(1)シングルプレイリスト、(2)シーケンシャルプレイリスト、(3)マルチプレイリスト、の3種類に大別することができる。

【0120】

(1)のシングルプレイリストは、図23Aに一例が示されるように、単一のプレイリストで構成されるシナリオである。タイムラインが定義可能で、シナリオ再生中の割り込みが無い。コンテンツが映画であれば、例えばディスクのローディング後に映画本編だけが再生されるようなシナリオである。

【0121】

(2)のシーケンシャルプレイリストは、図23Bに一例が示されるように、複数のプレイリストが分岐無しに直列的に並べられて構成されるシナリオである。各プレイリストは、プレイリストの末尾と次のプレイリストの先頭とが接続されるように並べられる。各プレイリストを通じてのタイムラインが定義可能である。コンテンツが映画であれば、例えばメニュー画面と映画本編との組み合わせからなるシナリオである。ディスクのローディング後にメニュー画面を表示させるプレイリストが実行され、メニュー画面に対して映画本編の再生が指示されると、次のプレイリストが実行され、映画の本編が再生される。

【0122】

(3)のマルチプレイリストは、図23Cに一例が示されるように、プレイリストの分岐や収束による接続を含むシナリオである。プレイリストを通じてのタイムラインを定義することができず、例えば各プレイリスト内にタイムラインが定義される。マルチプレイリストによれば、ユーザ入力によって再生内容が変化するようなインタラクティブ性やゲーム性を実現することができる。コンテンツが映画の場合には、例えば同一シーンを様々

10

20

30

40

50

なアングルで収録し、再生時にユーザの指示により選択して鑑賞できるようにしたマルチアングルを実現するシナリオである。

【0123】

再生専用メディアの場合、シナリオは、後述するように、HDMVディレクトリに一つ定義されるものとする。ユーザに対しては、シナリオをより細分化した単位を見せる必要がある。但し、プレイリストの単位がユーザの認識する単位と一致するとは限らない。例えば、一つのプレイリストに3本の映画が収録されているときには、各映画の頭出し点をユーザに見せる必要がある。このように、プレイリストの構造とは独立した頭出し点(エントリポイント)を、以下では、タイトルおよび/またはチャプタと呼ぶことにする。

【0124】

図24を用いてタイトルおよびチャプタについて説明する。タイトルは、シナリオ中の任意の再生開始点を表す。この図24の例では、プレイリスト470Aの先頭にタイトル1に設定されている。タイトル2は、プレイリスト470Dの途中に設定されている。プレイリスト470Aの先頭からタイトル2の直前までがタイトル1とされる。チャプタは、タイトルをさらに細分化した単位で、これもユーザが再生開始点として認識できる単位となる。タイトル1がチャプタによりさらに細分化される。図24の例では、タイトル1に対してチャプタ1、2および3が設定され、タイトル1が3つに細分化されている。また、図24に示されるように、タイトルおよびチャプタの位置は、プレイリストの途中を指定することもできる。

【0125】

次に、シナリオ記述に従って動作する再生装置のモデルを考える。以下では、このモデル化された再生装置をBD(Blu-ray disc)仮想プレーヤと称し、このBD仮想プレーヤの構成の定義をBD仮想プレーヤモデルと称する。

【0126】

図25を用いて、BD仮想プレーヤモデルについて説明する。BD仮想プレーヤ30は、ディスクのローディング後、この発明で定義したシナリオ記述言語により記述されディスクに記録されたシナリオを、PBCプログラム40として読み込み、シナリオの記述に従い動作する。

【0127】

BD仮想プレーヤ30は、この発明の実施の一形態により定義されるディスク状記録媒体を再生するもので、パーソナルコンピュータなどのコンピュータ環境上のオブジェクトである。コンピュータ環境は、汎用的なパーソナルコンピュータに限られず、例えば、この発明の実施の一形態により定義されるディスク状記録媒体などを再生するように専用の構成された再生装置および/または記録再生装置に組み込まれたソフトウェア環境も含む。なお、以下、この発明の実施の一形態により定義されるディスク状記録媒体を、ディスクと略称する。

【0128】

BD仮想プレーヤ30の状態は、概略的には、プレイリストおよびグラフィクスを再生する状態Aと、再生が停止された状態Bの2つの状態に分類できる。ある状態から別の状態への遷移や、ある状態の中での次の動作の指定は、BD仮想プレーヤ30のオブジェクトに対するコマンドにより行われる。

【0129】

なお、状態Aは、複数の動作を含む。状態Aに含まれる動作としては、高速再生、逆転再生といった変速再生や、ディスク上の任意の時刻から再生する飛び込み再生といった特殊再生などが考えられる。BD仮想プレーヤ30においてグラフィクスプレーン12による表示がなされているときは、これら変速再生や特殊再生が制限されることがある。

【0130】

PBC(Play Back Control)プログラム40は、例えばディスクに記録されているシナリオに対応する。詳細は後述するが、シナリオは、ディスクに記録されたプレイリストの再生方法や、メニュー画面の表示を指示する。PBCプログラム40とBD仮想プレーヤ

10

20

30

40

50

30とは、API(Application Programming Interface) 41を介してメソッドのやりとりを行い、ディスクに記録されたプレイリストを再生する。

【0131】

シナリオには、プレーヤの動作を指示するコマンドが並んだプログラムを含むコマンドを配置する領域として、グローバルコマンド領域と、ローカルコマンド領域の二つがある。

【0132】

グローバルコマンド領域は、シナリオ全体に有効なプログラムが格納されており、例えば、ディスクがプレーヤに挿入されたときに最初に自動的にプレーヤによって実行されて、パラメータの初期化を行い、メニュー画面を構成するプレイリストにジャンプするようなプログラムを記述する領域である。ローカルコマンド領域は、各プレイリストに関係するプログラムを記述する領域である。ローカルコマンドは、さらにプレコマンド、プレイアイテムコマンド、ポストコマンド、ボタンコマンドの4種類に分類される。

10

【0133】

図26は、この発明の実施の一形態におけるシナリオで記述されるコマンドによるBD仮想プレーヤ30の動作を概略的に示す。図26Aは、ディスクのローディング時の動作の例を示す。上述したように、シナリオは、後述するHDMVディレクトリに対して一つが作られる。ディスクがプレーヤに挿入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)、レジスタすなわち共通パラメータ32が初期化される(ステップS31)。そして、次のステップS32で、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、ディスク挿入時のように、ディスクの再生が初めて行われることをいう。

20

【0134】

この、ディスクローディング時に最初に読み込まれ、実行されるコマンド群(プログラム)を、グローバルコマンドと称する。グローバルコマンドは、例えば宣伝用映像(トレーラー)やメニュー画面を構成するプレイリストへのジャンプ命令が記述されており、その命令通りにプレーヤがプレイリストを再生していくことになる。

【0135】

図26Bは、プレーヤ30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指示された場合の動作の例を示す。これは、上述の図25における状態Bから状態Aへの状態遷移に相当する。最初の停止状態(ステップS40)に対して、ユーザにより、例えばリモコンなどを用いて再生が指示される(UOP: User Operation)。再生が指示されると、まず、レジスタすなわち共通パラメータ32が初期化され(ステップS41)、次のステップS42で、プレイリストの再生フェイズに移行する。

30

【0136】

プレイリストの再生フェイズにおけるプレイリストの再生について、図27を用いて説明する。図27Aは、プレイリストが単一のプレイアイテムからなる場合の例である。プレイリストは、プレコマンド領域、プレイアイテムコマンド領域およびポストコマンド領域の3箇所にプログラムが配される。プレイリストの再生フェイズに移行すると、まずプレコマンド領域のプレコマンドが実行され(ステップS10)、プレコマンドの実行が終了されると、プレイリストを構成するプレイアイテムの再生フェイズに移行する(ステップS11)。プレイアイテムの再生フェイズでは、プレイアイテムにおいて開始点および終了点で指定されたストリームが再生される(ステップS110)。終了点までの再生が終了した時点で、プレイアイテムコマンドが実行される(ステップS111)。プレイアイテムコマンドが実行されると、次にポストコマンド領域のポストコマンドが実行され(ステップS12)、プレイリストの再生が終了される。

40

【0137】

なお、ポストコマンドは、一般的な使用においては、次に再生すべきプレイリストへのジャンプ命令や、メニュー画面を構成するプレイリストへのジャンプ命令が記述される。ジャンプ命令が無い場合には、そこで再生が停止し、プレーヤは、停止状態(図25の状

50

態 B) に遷移される。

【 0 1 3 8 】

図 2 7 B は、プレイリストが複数のプレイアイテムを含む場合の例である。この場合でも、プレイリスト中にプレコマンド領域、プレイアイテムコマンド領域およびポストコマンド領域が配され、それぞれプログラムが格納される。複数のプレイアイテムを含む場合、プレイアイテムコマンド領域では、プレイアイテム数分のプレイアイテムストリームおよびプレイアイテムコマンドが、時系列に従って配置される。

【 0 1 3 9 】

プレイリストが複数のプレイアイテムを含む場合でも、プレイリストの再生フェイズに移行すると、まず、プレコマンドが実行される (ステップ S 1 0)。次のプレイアイテムの再生フェイズでは、プレイアイテムの開始点から終了点までのストリームの再生と、プレイアイテムコマンドの実行とが、プレイリストが含むプレイアイテムの数だけ実行される。図 2 7 B の例では、最初のプレイアイテムストリームが再生され (ステップ S 1 1 0 - 1)、対応するプレイアイテムコマンドが実行される (ステップ S 1 1 1 - 1)。次に、図示は省略するが、2 番目のプレイアイテムストリームが再生され (ステップ S 1 1 0 - 2)、対応するプレイアイテムコマンドが実行される (ステップ S 1 1 1 - 2)。これをプレイアイテムの数だけ行い、最後のプレイアイテムストリームが再生され (ステップ S 1 1 0 - n)、対応するプレイアイテムコマンドが実行されると (ステップ S 1 1 1 - n)、プレイアイテムの再生フェイズが終了する。プレイアイテムの再生フェイズが終了すると、ポストコマンドが実行され (ステップ S 1 2)、このプレイリストの再生フェイズが終了される。

10

20

【 0 1 4 0 】

図 2 8 は、コマンドの一部を示す。この図 2 8 に例示されるようなコマンドを、ポストコマンド領域や後述するボタンコマンド領域に書くことにより、特定のプレイリストへのジャンプを実現することができる。なお、コマンドは、この図 2 8 に例示されるに止まらず、さらに他のコマンドを定義することも可能である。

【 0 1 4 1 】

再生開始位置指定に関するコマンドについて説明する。コマンド LinkPlayList (playListNumber) で、"playListNumber" で指定されたプレイリストの再生開始が指示される。コマンド LinkPlayItem (playListNumber, playItemNumber) で、指定のプレイリストの指定のプレイアイテムからの再生開始が指示される。"playItemNumber" は、"PlayItem_id" であり、値が「 0 」から始まる。"playItemNumber" に値「 0 」を指定することで、当該プレイアイテムが属するプレイリストの先頭から再生される。

30

【 0 1 4 2 】

コマンド Link (position) (object) で、シナリオ内での移動が指示される。すなわち、現在移動中の箇所から前後のプレイリスト、プレイアイテムおよびチャプタに移動することがこのコマンドで指示される。なお、パラメータ "position" は、"prev"、"next"、"top"、"Parent" または "tail" の何れかの値を取り、パラメータ "object" で示される移動対象 (プレイリスト、プレイアイテムまたはチャプタ) に対する移動方法が指示される。

40

【 0 1 4 3 】

コマンド Exit で、シナリオ再生の停止が指示される。この場合、標準レジスタの値は、保持されない。コマンド RSM で、プレーヤ内のメモリに保存されているレジューム情報を読み出し、レジスタにセットして再生を開始する。

【 0 1 4 4 】

プレーヤ状態の取得に関するコマンドについて説明する。コマンド getMenuDescription Language () で、メニュー表示の際に用いられる言語が取得される。コマンド getScenarioNumber ()、コマンド getPlayListNumber () およびコマンド getChapterNumber () で、再生中のシナリオ番号、プレイリスト番号およびチャプタ番号がそれぞれ取得される。コマンド getPlayerSupport () で、プレーヤのバージョン情報が取得される。

50

【 0 1 4 5 】

ビデオストリームに関するコマンドについて説明する。コマンドgetVideoStreamAvailability()で、指定のビデオストリームが含まれているか否かが取得される。コマンドsetVideoStreamNumber()で、デコードするビデオストリームが指定される。コマンドgetVideoStreamNumber()で、選択中のビデオストリームの番号が取得される。コマンドgetVideoStreamAttr()で、選択中のビデオストリームの属性が取得される。なお、ビデオストリームの属性は、例えば、そのビデオストリームの符号化方式、解像度、アスペクト比、アスペクト比が4:3のときのディスプレイモード、クローズドキャプションの有無である。コマンドsetAngleNumber()で、アングル番号が指定される。コマンドgetAngleNumber()で、選択中のアングル番号が取得される。コマンドgetMaxVideoStreams()で、ビデオストリームの最大数が取得される。

10

【0146】

次に、この発明の実施の一形態によるシナリオを記述するコマンドやデータベースの、ディスクへの格納方法について説明する。図29は、この実施の一形態による一例のファイルの管理構造を示す。ディスク上には、先ず、1つのルートディレクトリが作成される。このルート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。

【0147】

ルートディレクトリの下に、ディレクトリBDMVが置かれる。なお、図示はされていないが、ディレクトリBDMVは、上述の図4の如く、ルートディレクトリの下に複数置くことができる。なお、以下では、この発明の実施の一形態で定義される方式をBDMVと総称する。

20

【0148】

ディレクトリBDMVの下には、2つのファイルすなわちファイル「scenario.hdmv」およびファイル「entrylist.data」が置かれると共に、ディレクトリ「PLAYLIST」、ディレクトリ「CLIPINF」、ディレクトリ「STREAM」といった複数のディレクトリが置かれる。

【0149】

図30は、ファイル「scenario.hdmv」の一例の構造を表すシンタクスを示す。このファイル「scenario.hdmv」は、ディスク挿入時などのイニシャルアクセス時に最初に読み込まれ実行されるファイルである。ファイル「scenario.hdmv」は、最初にファイル識別記号(フィールドtype_indicator)、バージョン番号(フィールドversion_number)が記述され、その後、機能毎のデータを集めたブロックが並んでいる。

30

【0150】

フィールドtype_indicatorは、32ビットのデータ長を有し、このファイルがファイル「scenario.hdmv」であることを表す特定の文字列が格納されている。フィールドversion_numberは、32ビットのデータ長を有し、バージョン番号が入る。フィールドScenario_start_addressは、32ビットのデータ長を有する符号無し整数で、ブロックScenario()が開始される位置をファイル「scenario.hdmv」の先頭からの相対バイト数で表した値が格納される。

【0151】

ブロックAutoplay()は、ファイルの先頭から41バイト目の固定位置から始まる。ブロックAutoplay()には、上述した、イニシャルアクセス時(ディスク挿入時のようにディスクの再生が始めて行われる時)に実行されるプログラムが記述される。上述したグローバルコマンドは、このブロックAutoplay()に格納される。Autoplay()の後には、任意の数のパディングワード(padding_word)が入り、ブロックの後に空きを持たせることができるようになっている。

40

【0152】

図31は、図30におけるブロックAutoplay()の一例のデータ構造を表すシンタクスを示す。フィールドlengthは、データ長が32ビットの符号無し整数で、このフィールドlengthの直後からブロックAutoplay()の終わりまでのデータ長を、バイトで表したものが記述される。フィールドnumber_of_commandsは、その後続くフィールドcommand(i)の数

50

を表す。フィールドcommand(i)は、データ長が32ビットの値で、上述の図28に一例が記されている、プレイヤーのパラメータのセット、指定のプレイリストの再生開始、演算などの命令が記述されている。

【0153】

ブロックScenario()は、これまで説明してきた「シナリオ」が記述される。プレイリストの再生順序の情報およびプレイリスト毎に持つローカルコマンド領域などが、このブロックScenario()に記述される。

【0154】

図32は、ブロックScenario()の一例の構造を表すシンタクスを示す。ブロックScenario()は、シナリオの情報、すなわちプレイリスト間のリンクを定義するブロックであって、上述したプレコマンド、ポストコマンド、プレイアイテムコマンドなどの情報およびコマンドそのものが格納される。ブロックScenario()は、ブロックScenario()内に格納されるコマンドの情報を示すフィールドが記述される領域と、実際のコマンドが列挙される領域とから構成される。

10

【0155】

フィールドlengthは、このフィールドlengthの直後からブロックScenario()の終わりまでの長さをバイトで表現した値が記述される。フィールドnumber_of_PlayListsは、シナリオを構成するプレイリストの数を表す。このフィールドnumber_of_PlayLists以降は、プレイリスト毎に持つデータである。すなわち、フィールドnumber_of_PlayListsの次行からのforループにより、フィールドnumber_of_PlayListsで示される値をループカウンタiの最大値として、プレイリスト毎のデータが記述される。

20

【0156】

フィールドPre_Command_start_idは、プレイリストを再生する前に実行されるプログラム(プレコマンド)の、コマンドテーブル中での開始番号を表す。すなわち、フィールドPre_Command_start_idにより示される番号は、後述するフィールドPL_Command(i)が列挙されるforループ内でのループカウンタjを表す。同様に、フィールドPost_Command_start_idは、プレイリストを再生した後に実行されるプログラム(ポストコマンド)の、コマンドテーブル中での開始番号を表す。すなわち、フィールドPost_Command_start_idにより示される番号は、後述するフィールドPL_Command(j)が列挙されるforループ内でのループカウンタjを表す。

30

【0157】

フィールドnumber_of_Pre_Commandsは、このプレイリストを再生する前に実行されるプログラム(プレコマンド)を構成するコマンドの数を表す。同様に、フィールドnumber_of_Post_Commandsは、このプレイリストを再生した後に実行されるプログラム(ポストコマンド)を構成するコマンドの数を表す。これらのプログラムは、後述するコマンドテーブル内に記述される。

【0158】

次のフィールドnumber_of_PlayItemsは、このプレイリストを構成するプレイアイテムの数を表す。フィールドPI_Command_start_idは、プレイアイテムの再生後に実行するコマンドの、コマンドテーブル中の開始番号を表す。すなわち、フィールドPI_Command_start_idにより示される番号は、後述するコマンドテーブル内でのループカウンタjを表す。フィールドnumber_of_PI_Commandsは、プレイアイテムの再生後に実行するコマンド(プレイアイテムコマンド)の数を表す。すなわち、後述するコマンドテーブル内において、フィールドPI_Command_start_idで示される位置から始まり、フィールドnumber_of_PI_Commandsで表される数のコマンドを、プレイアイテムの再生後に実行する。

40

【0159】

フィールドnumber_of_PL_Commandsは、次行からのコマンドテーブル中のコマンドの数を表す。コマンドテーブルは、フィールドPL_Command(j)が列挙されるforループにより構成される。コマンドテーブル内のコマンドには、順に番号jが割り当てられている。番号jは、コマンドテーブルを記述するためのforループのループカウンタjに対応する。

50

フィールドPL_Command(j)は、一つのコマンドを表し、番号jは、上述したフィールドPre_Command_start_id、フィールドPost_Command_start_idおよびフィールドPI_Command_start_idから参照される。

【0160】

図33は、ファイル「entrylist.data」の一例のデータ構造を表したシンタクスを示す。ファイル「entrylist.data」は、最初にファイル識別記号(フィールドtype_indicator)、バージョン番号(フィールドversion_number)、ブロックの開始アドレス(フィールドScenarioEntry_start_address)が記述され、その後、機能毎のデータを集めたブロックが並んでいる。

【0161】

フィールドtype_indicatorは、データ長が32ビットを有し、このファイルがタイトルやメニューのエントリーポイントを記述したファイルであることを表す特定の文字列が格納されている。フィールドversion_numberは、データ長が32ビットを有し、バージョン番号が格納される。フィールドScenarioEntry_start_addressは、データ長が32ビットの符号無しの整数で、ブロックScenarioEntry()が開始される位置を、ファイル「entrylist.data」の先頭からの相対バイト数で表した値が格納される。

【0162】

図34は、ブロックAppInfo()の一例の構造を表すシンタクスを示す。フィールドlengthは、データ長が32ビットの符号無しの整数で、このフィールドlengthの直後からブロックAppInfo()の終わりまでの長さをバイトで表したものが記述される。フィールドHDMV_name_character_setは、後述するフィールドHDMV_nameを記述する文字セットを表す。フィールドPIN_valid_flagは、再生時に暗証番号を設定するか否かを表し、設定有効の場合は、直後のフィールドPINが暗証番号を表す。フィールドHDMV_name_lengthは、次のフィールドHDMV_nameの有効部分の長さを表す。フィールドHDMV_nameは、このファイル「entrylist.data」が置かれるディレクトリHDMVに付けられた名称がテキスト形式で格納されている領域である。フィールドHDMV_nameは、データ長が255バイトの固定長とされる。実際に名称が格納される部分は、フィールドHDMV_nameの先頭からフィールドHDMV_name_lengthで指定されている長さまでとされる。

【0163】

図35は、ブロックScenarioEntry()の一例の構造を表すシンタクスを示す。ブロックScenarioEntry()は、シナリオ内の頭出し点が羅列される。ここで、シナリオは、上述したようにディレクトリHDMVに対して一つ作られ、ディレクトリHDMVの下に置かれる複数のプレイリストを結び付けて再生順序を定義する。一方、ユーザから見た場合、シナリオは、必ずしも一つの映像・音声単位として見える訳ではなく、複数の「タイトル」から構成されているように見える。

【0164】

例えば、一枚のディスクに映画が3本収録されている場合、再生順序を定義するシナリオは、ディスクに一つのみ存在するが、ユーザには当該ディスクに3つのタイトルが収録されているように見える。あるいは、その3つのタイトルのリストを表示し、ユーザに選択させるタイトルメニュー画面も含めて、4つのタイトルに分割されているように見える。なお、メニュー画面もユーザにとっては一まとまりの映像、音声単位であるので、この発明の実施の一形態では、タイトルの一種としてみなすことにする。

【0165】

このように、プレイリストの結びつきを定義するシナリオという単位と、ユーザが一まとまりの映像、音声として認識する単位には違いがあるため、上述のように、シナリオ内に頭出し点を定義しておく必要がある。このシナリオ内の頭出し点を、タイトルエントリーと称する。このブロックScenarioEntry()に、タイトルエントリーの情報が記述される。

【0166】

図35の説明に戻り、フィールドlengthは、データ長が32ビットの符号無しの整数で、このフィールドlengthの直後からブロックScenarioEntry()の終わりまでの長さをバイ

10

20

30

40

50

トで表したものが格納される。フィールドname_character_setは、後述するフィールドToPMenu_nameとフィールドTitle_nameとを表現する文字セットを表す。

【0167】

次のブロックTop Menu PL()は、ユーザがリモコンのタイトルメニューキーを押したときに表示されるメニューを構成しているプレイリストあるいはプレイリスト群へのエン트리ポイントを指定する。トップメニューは、シナリオに一つあり、例えばタイトルをユーザに提示するためのメニューとして利用される。トップメニューに対して、さらにオーディオや字幕を設定するサブメニューを下層メニューとして設けることができる。サブメニューは、ストリーム設定メニューとも称される。

【0168】

フィールドflagsは、詳細は省略するが、トップメニューに関する属性情報を表す領域である。フィールドTopMenu_ref_to_PlayList_file_nameは、トップメニューを構成するプレイリストあるいはプレイリスト群の入り口となるプレイリストを指定する。フィールドTopMenu_ref_to_PlayItem_idは、フィールドTopMenu_ref_to_PlayList_file_nameで指定したプレイリスト中の特定のプレイアイテムからトップメニューが開始されている場合、そのプレイアイテムの番号を指定する。プレイリストの先頭からの再生であれば、フィールドTopMenu_ref_to_PlayItem_idの値は、「0」となる。フィールドTopMenu_name_lengthは、トップメニューに付された名前の長さを表す。フィールドTopMenu_nameは、トップメニューに付された名前の文字列が格納される。

【0169】

ブロックTop Menu PL()の次に、タイトルに関する情報が記述される。フィールドnumber_of_Titlesは、その直後のforループ内のタイトル頭出し点(タイトルエントリ)の数を表す。フィールドflagsは、詳細は省略するが、タイトルに関する属性情報を表す領域である。フィールドTitle_ref_to_PlayList_file_nameは、タイトルエントリを含むプレイリストのファイル名を表す。フィールドTitle_ref_to_PlayItem_idは、フィールドTitle_ref_to_PlayList_file_nameで指定したプレイリストの中の特定のプレイアイテムからタイトルが始まる場合に用いる。フィールドTitle_name_lengthは、タイトルに付された名前の長さを表す。フィールドTitle_nameは、タイトルに付された名前の文字列が格納される。

【0170】

次に、サブメニューに関する情報が記述される。「Stream Setup Menu」以下に、プレイアイテム毎に持つことができるストリーム設定メニュー(すなわちサブメニュー)を構成するプレイリストあるいはプレイリスト群へのエントリポイントが記述されている。ストリーム設定メニューは、オーディオ、字幕、アングルなどの切り替えなど、プレイリスト毎に個別にメニューを用意したい場合に利用することができる。例えば、上述の図20におけるボタン64、65を押すことで表示される画面がサブメニューである。

【0171】

フィールドnumber_of_PlayListsは、ストリーム設定メニューに用いられるプレイリストの数を表す。このフィールドnumber_of_PlayListsの値が直後のforループのループ回数として用いられる。フィールドSSMenu_flagsは、詳細は省略するが、ストリーム設定メニューに関連する属性情報を格納する領域である。フィールドSSMenu_ref_to_PlayList_file_nameは、ストリーム設定メニューを構成するプレイリストあるいはプレイリスト群の入り口となるプレイリストを指定する。フィールドSSMenu_ref_to_PlayItem_idは、フィールドSSMenu_ref_to_PlayList_file_nameで指定したプレイリスト中の特定のプレイアイテムからストリーム設定メニューが開始されている場合、そのプレイアイテムの番号を指定する。プレイリストの先頭からの再生であれば、フィールドSSMenu_ref_to_PlayItem_idの値は、「0」となる

【0172】

図36は、ファイル「xxxxx.mpls」の一例の構造を表すシンタクスを示す。図36において、ファイル「xxxxx.mpls」の内部は、機能別の情報毎にブロックが構成される。フィ

10

20

30

40

50

ールドtype_indicatorにこのファイルを示す文字列が格納され、フィールドversion_numberにこのファイルのバージョンが示される。また、フィールドPlayList_start_addressおよびPlayListMark_start_addressには、それぞれ対応するブロックの先頭アドレスが例えばデータ長が32ビットのアドレス情報として示される。

【0173】

ブロックPLControlInfo()は、このプレイリストに関する属性情報が格納される。ブロックPlayList()は、このプレイリストを構成するプレイアイテムに関する情報が格納される。ブロックPlayListMark()は、このプレイリストに付されるマークの情報が格納される。

【0174】

ファイル「xxxxx.mpls」において、ブロックPLControlInfo()、PlayList()およびPlayListMark()の先頭アドレスがこれらのブロックより前に示されるために、各ブロックの前および/または後ろに、任意の長さのパディングデータpadding_wordを挿入することができる。但し、ファイル「xxxxx.mpls」の最初のブロックであるブロックPLControlInfo()の開始位置は、ファイルの先頭から41バイト目に固定される。

【0175】

図37は、ブロックPLControlInfo()の一例の構造を表すシンタクスを示す。このブロックPLControlInfo()は、プレイリストの再生において直接的に必要とされない、プレイリストに関する各種の属性情報が格納される。フィールドPlayList_character_setは、プレイリストに関する文字列情報の文字セットが指定される。

【0176】

フィールドPL_playback_typeは、図38に一例が示されるような値を取り、このプレイリストがシーケンシャルに再生される通常のプレイリストか、プレイアイテムをランダム再生するプレイリストか、プレイアイテムをシャッフル再生するプレイリストであるかを表す。ランダムシャッフルは、プレイリストの単位で指定する。すなわち、一つのプレイリストに通常再生のプレイアイテムとランダムシャッフルのプレイアイテムブロックを混在させない。再生専用の記録媒体の場合には、制作者の意図により、ランダム再生やシャッフル再生を指定されることがあるため、このような情報が必要とされる。

【0177】

フィールドplayback_countは、このプレイリストがランダム再生あるいはシャッフル再生のプレイリストであるとき、プレイアイテムの再生回数を指定する。すなわち、このフィールドplayback_countにより、ランダム再生あるいはシャッフル再生に用いるプレイアイテムの数を指定できる。

【0178】

フィールドPL_UOP_mask_table()は、ユーザ操作を制限する情報が格納される。再生、早送り、早戻しなどのユーザ操作を、このプレイリストの再生中に禁止したい場合、この領域で適切な指定をする。例えば、このフィールドPL_UOP_mask_table()内の値を所定に設定することで、警告表示や著作権の表示を早送りなどで飛ばして再生されないようにできる。

【0179】

フィールドPL_random_access_modeは、図39に一例が示される値を取り、このプレイリスト中の任意の箇所に飛び込み再生をするランダムアクセスが可能か否かを示す。例えば、必ずユーザに見せたいプレイリストがある場合に、フィールドPL_random_access_modeの値を{0x1}に設定する。これにより、このプレイリストを再生する際には、早送り、早戻し、任意時刻からの再生などが不可能になる。再生専用の記録媒体においては、制作会社のロゴ表示や注意事項など、ユーザに必ず見せたいシーンが収録されている場合がある。このフィールドPL_random_access_modeは、そのようなシーンを変速再生等で飛ばされないようにするために必要な情報である。

【0180】

フィールドPlayList_durationは、プレイリストの再生時間を示す。フィールドPlayLis

10

20

30

40

50

t_nameは、フィールドPlaylist_name_lengthに示される値を有効長としてプレイリスト名が示される。フィールドPlaylist_detailは、フィールドPlaylist_detail_lengthに示される値を有効長としてプレイリストの詳細情報が示される。

【0181】

図40は、ブロックPlaylist()の一例の構造を表すシンタクスを示す。フィールドlengthは、フィールドlengthの直後のフィールドからこのブロックPlaylist()の終端までのバイト長を示す。フィールドnumber_of_PlayItemsは、このプレイリストを構成するプレイアイテムの数を示す。フィールドnumber_of_SubPlayItemsは、このメインのプレイアイテムと同時に再生される補助的なプレイアイテム(サブプレイアイテム)の数を示す。

【0182】

ブロックPlayItem()は、プレイアイテムの情報が格納される。ブロックSubPlayItem()は、サブプレイアイテムの情報が格納される。

【0183】

図41は、ブロックPlayItem()の一例の構造を表すシンタクスを示す。フィールドClip_Information_file_nameは、このプレイアイテムが参照しているクリップと1対1に対応するクリップ情報ファイル(拡張子が「clpi」であるファイル)のファイル名が文字列で示される。

【0184】

フィールドClip_codec_identifierは、このプレイアイテムにより参照されるクリップの符号化方式を示す。この実施の一形態では、フィールドClip_codec_Identifierは、値「M2TS」に固定的とされる。すなわち、この実施の一形態では、プレイアイテムにより参照されるクリップの符号化方式が値「M2TS」により示される方式に固定的とされる。

【0185】

フラグis_multi_angleは、このプレイアイテムが後述するマルチアングル構造であるかどうかを表す(第1のフラグ)。

【0186】

フィールドconnection_conditionは、このプレイアイテムと次のプレイアイテムとがどのように接続されているかを示す情報である。すなわち、フィールドconnection_conditionにより、プレイアイテムとプレイアイテムとの間が継ぎ目なくシームレスに再生できるか否かが示される。

【0187】

フィールドref_to_STC_idは、このプレイアイテムにより参照されるクリップ内のシーケンスSTC_sequenceを指定する。シーケンスSTC_sequence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Blu-ray Disc規格独自の構造である。シーケンスSTC_sequence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STC_sequence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STC_sequenceに存在していなければならない。フィールドref_to_STC_idでは、番号STC_idによりシーケンスSTC_sequenceが指定される。

【0188】

フィールドIN_timeおよびOUT_Timeは、このプレイアイテムにおける開始点および終了点の、シーケンスSTC_sequence上でのタイムスタンプpts(presentation_time_stamp)をそれぞれ示す。

【0189】

フィールドPI_UOP_mask_table()は、ユーザ操作を制限するためのデータが格納される。ここで制限されたユーザ操作は、仮にユーザがその操作を行ったとしても、プレーヤは反応してはならない。例えばメニュー画面で早送りが実行されないようにする場合に、ここで早送りのユーザ操作を無効にするような設定をしておく。

10

20

30

40

50

【 0 1 9 0 】

なお、このプレイアイテム毎に設けられるフィールドPI_UOP_mask_table()は、上述したプレイリストの再生に関するブロックPLControlInfo()に設けられるフィールドPL_UOP_mask_table()と同様の目的の情報が格納される。プレイリストとプレイアイテムとの何方かで禁止となっていれば、そのユーザ操作は禁止となる。すなわち、プレイリストの情報と、プレイアイテムの情報とで和演算がなされ、あるプレイアイテム再生中でのユーザ操作の制限が決まることになる。

【 0 1 9 1 】

フィールドPID_filter()は、詳細は省略するが、このプレイアイテムで再生するストリームの優先順位を決めるテーブルである。

10

【 0 1 9 2 】

フィールドPI_random_access_modeは、図 4 2 に一例が示されるような値を取り、このプレイアイテム中の任意の箇所に飛び込み再生をするランダムアクセスが可能か否かを示す。例えば、必ずユーザに見せたいプレイリストがある場合に、フィールドPI_random_access_modeの値を〔 0 x 1 〕に設定しておく。これにより、このプレイアイテムを再生する際に、早送り、早戻し、任意時刻からの再生などを不可能にできる。

【 0 1 9 3 】

フィールドstill_modeは、このプレイアイテム再生後に一時停止を行うかを決定するフィールドである。フィールドstill_modeは、図 4 3 に一例が示されるような値を取り、例えばフィールドstill_modeの値が〔 0 x 1 〕のときは一時停止することを表し、次のフィールドstill_timeのフィールドで指定された時間、一時停止を行う。これにより、スライドショーなどのような、静止画像を一定間隔で次々と表示するような再生を実現することができる。この場合には、静止画 1 枚ずつがそれぞれプレイアイテムとなっている。なお、フィールドstill_timeは、有限の時間だけでなく、ユーザが入力するまで一時停止し続けるという設定（ポーズ設定）も可能となっている。例えばフィールドstill_modeの値を〔 0 x 2 〕とすることで、ポーズ設定が可能とされる。

20

【 0 1 9 4 】

上述したフラグis_multi_angleの値が例えば「 1 」とされていれば、このプレイアイテムがマルチアングルであるとされ、「Angle」以下に、マルチアングルのための情報が追加される。

30

【 0 1 9 5 】

フィールドnumber_of_anglesは、アングル数を表す。フィールドis_seamless_angle_changeは、図 4 4 に一例が示されるような値を取り、アングルがシームレスに切り替え可能なように、すなわち、アングル間をなめらかに切り替えることができる状態に各アングルがディスク上配置されているか否かを表す。

【 0 1 9 6 】

次のforループは、アングルを構成するクリップの情報が記述される。forループ中のフィールドClip_Information_file_nameは、このプレイアイテムが参照しているクリップと 1 対 1 に対応するクリップ情報ファイル（拡張子が「clpi」であるファイル）のファイル名が文字列で示される。フィールドref_to_STC_idは、このプレイアイテムにより参照されるクリップ内のシーケンスSTC_sequenceを指定する。

40

【 0 1 9 7 】

ここで、値angle_id=0に相当するアングルは、アングルを構成しない通常のプレイアイテムと同じように、このブロックPlayItem()の前半部分で既に定義されている。値angle_id=1以降のアングルがこのforループの中で定義されている。したがって、このforループの中には、値angle=0に相当するアングルは、含まれていない。

【 0 1 9 8 】

図 4 5 は、ブロックSubPlayItem()の一例の構造を表すシンタクスを示す。フィールドlengthは、この直後のフィールドからSubPlayItem()の終わりまでの長さをバイトで表したものである。フィールドClip_Information_file_nameは、このサブプレイアイテムが参照

50

しているクリップと1対1に対応するクリップ情報ファイル(拡張子が「clpi」であるファイル)のファイル名が文字列で示される。

【0199】

フィールドClip_codec_identifierは、このサブプレイアイテムにより参照されるクリップの符号化方式を示す。この実施の一形態では、フィールドClip_codec_identifierは、値「M2TS」に固定的とされる。

【0200】

フィールドis_repeat_flagは、図46に一例が示されるような値を取り、このサブプレイアイテムをメインのプレイアイテム(メインパス)と非同期に繰り返し再生するか否かを表すフラグである。例えば、このフィールドis_repeat_flagの値が「1」の場合は、メインのプレイアイテムの再生が終了するまで、メインのプレイアイテムとは非同期に、このサブプレイアイテムが繰り返し再生される。このフィールドis_repeat_flagの値が「0」の場合は、このサブプレイアイテムは、メインのプレイアイテムと同期して、一度だけ再生される。

10

【0201】

例えば、このサブプレイアイテムがオーディオのみのサブプレイアイテムである場合、フィールドis_repeat_flagの値を「1」とすることで、BGM(Back Ground Music)再生を行うことができる。

【0202】

フィールドSubPlayItem_typeは、このサブプレイアイテムがどのような性質を持ったサブプレイアイテムであるかを表す。例えば、フィールドSubPlayItem_typeの値が「1」であれば、オーディオのみのサブプレイアイテムであることを表す。

20

【0203】

フィールドref_to_STC_idは、このプレイアイテムにより参照されるクリップ内のシーケンスSTC_sequenceを指定する。フィールドSubPlayItem_IN_timeおよびSubPlayItem_OUT_timeは、このサブプレイアイテムにおける開始点および終了点の、シーケンスSTC_sequence 上でのタイムスタンプpts(presentation_time_stamp)をそれぞれ示す。

【0204】

上述のフィールドis_repeat_flagの値が例えば「0」であって、このサブプレイアイテムがメインのプレイアイテムと同期して再生されることを示す場合、フィールドsync_PlayItem_idとフィールドsync_start_PTS_of_PlayItemにより、サブプレイアイテムがメインのプレイアイテムのどの時刻から同期再生されるかが指定される。

30

【0205】

図47に一例が示されるように、フィールドsync_PlayItem_idで、メインパスのプレイアイテムを指定し(PlayItem=1)、フィールドsync_start_PTS_of_PlayItemで、サブプレイアイテムが再生開始されるメインのプレイアイテム上の時刻を表す(t1)。また、サブプレイアイテムとして、対応するクリップがフィールドSubPlayItem_IN_timeおよびフィールドSubPlayItem_OUT_timeで指定される期間、再生されることが示されている。

【0206】

図48は、ファイル「zzzzz.clpi」の一例の構造を表すシンタクスを示す。図48において、ファイル「zzzzz.clpi」の内部は、機能別の情報毎にブロックが構成される。フィールドtype_indicatorにこのファイルを示す文字列が格納され、フィールドversion_numberにこのファイルのバージョンが示される。また、フィールドSequenceInfo_start_address、フィールドProgramInfo_start_address、フィールドCPI_start_addressおよびフィールドClipMark_start_addressで、それぞれ対応するブロックの開始位置が示される。

40

【0207】

図49は、ブロックClipInfo()の一例の構造を表すシンタクスを示す。フィールドlengthは、この直後のフィールドからブロックClipInfo()の終わりまでの長さをバイトで表す。フィールドClip_stream_typeは、クリップAVストリームの種別を表す。再生専用の規格においては、通常のクリップであることを示す値、例えば値「1」に固定的としてよい

50

。

【0208】

フィールドapplication_typeは、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application_typeは、図50に一例が示されるような値を取り、このクリップAVストリームが通常のビデオストリームであるか、静止画の表示に適した多重化がなされているストリームであるかを表す。

【0209】

より具体的には、この例によれば、フィールドapplication_typeの値が「1」で、対応するクリップAVストリームのファイルがこの実施の一形態によるHDMVトランスポートストリームのルールに従っていることを示す。このとき、当該クリップAVストリームは、通常の動画が再生される。

10

【0210】

フィールドapplication_typeの値が「2」で、対応するクリップAVストリームのファイルがオーディオ再生に同期する静止画用の、HDMVトランスポートストリームのルールに従っていることを示す。このとき、当該クリップAVストリームは、例えばMPEG2形式のファイルであって、ビデオデータおよびオーディオデータがマルチプレクスされている。また、ビデオデータは、例えばMPEG2におけるIピクチャが静止画として並んでいるような構成とされる。これにより、例えばオーディオの時間軸に同期して表示されるスライドショーのような再生が可能とされる。このような再生を、タイムベーススライドショーと称する。

20

【0211】

フィールドapplication_typeの値が「3」で、対応するクリップAVストリームのファイルがオーディオとは非同期に再生される静止画用の、HDMVトランスポートストリームのルールに従っていることを示す。このとき、オーディオデータとビデオデータとは、別ファイルとされ、例えば、オーディオデータが再生されている間、ビデオデータは、静止画が任意の間隔またはユーザの指定に基づき切り換えられて表示される。ビデオデータは、上述と同様に、例えばMPEG2におけるIピクチャが静止画として並んでいるような構成とすることができる。このような再生を、ブラウザブルスライドショーと称する。

30

【0212】

なお、フィールドapplication_typeの値が「0」のときは、対応するクリップAVストリームがHDMVトランスポートストリームのルールに従っていない場合である。

【0213】

ここで、静止画の表示に適した多重化とは、主として静止画スライドショーのようなアプリケーションの実現を容易にすることを想定している。このようなアプリケーションにおいては、静止画1枚と、その上に重なる字幕やグラフィクスデータとをカプセル化して多重化すると、読み込みが容易となる。

【0214】

通常の動画像と同様の多重化をしてしまうと、ある静止画と同時に表示されるべき字幕が、前の静止画の画像データ付近に多重化されているような状況（いわゆる多重化位相差）が発生し、より広範囲のストリームデータを読みださないと字幕とグラフィクスとが重ねられた静止画像1枚を表示することができない。

40

【0215】

すなわち、この発明の実施の一形態においては、ビデオデータや字幕を表示するためのグラフィクスデータは、MPEGにおけるTS(Transport Stream)パケットに格納されて伝送される。1つのTSパケットは、188バイトからなり、上述のビデオデータやグラフィクスデータは、それぞれTSパケットに収まるように分割されてTSパケットに格納される。このとき、ある静止画像データ（画像P1とする）に対応した字幕データのパケットが、次以降の静止画像データ（画像P2とする）のパケットの後ろに配置されてしまうと、画像P1に対応した字幕を表示するために、画像P2の後ろまでデータを読み込んで

50

おかなければならない。

【0216】

ある静止画一枚とそれに付随する字幕およびグラフィックスだけを対象として多重化を行えば（カプセル化）、他のデータが混入しないストリームを作ることができる。それを静止画ごとに繰り返してストリームをつなげていくと、静止画（および当該静止画に付随する字幕、グラフィックスデータ）毎のデータが直列につながった一本のストリームが出来る。このような多重化を行ったストリームを、静止画用のHDMVストリームとしている。

【0217】

さらに、静止画用のHDMVストリームには、上述したように、タイムベーススライドショーと、ブラウザブルスライドショーとの2種類がある。この実施の一形態では、この2種類を、フィールドapplication_typeで別の番号を割り当てて、区別している。

【0218】

このように、静止画とそれに付随する字幕やグラフィックスとをカプセル化して記録することで、静止画を切り替えながら再生する際のアクセス性が向上する。

【0219】

図49の説明に戻り、フィールドTS_recording_rate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num_of_source_packetsは、クリップAVストリームに含まれるパケット数を表す。フィールドBD_system_useおよびブロックTS_type_info_block()は、この発明と関連が低いので説明を省略する。

【0220】

図51は、ブロックSequenceInfo()の一例の構造を表すシンタクスを示す。フィールドlengthは、この直後のフィールドからブロックSequenceInfo()の終わりまでの長さをバイトで表したものである。フィールドnum_of_ATC_sequencesは、連続した時間に記録されたことを表すシーケンスATC_sequenceの数を示す。再生専用の媒体の場合、シーケンスATC_sequenceの数は、「1」にすることができるため、詳細は省略する。フィールドSPN_ATC_startは、シーケンスATC_sequenceの開始をパケット番号で表したものであり、シーケンスATC_sequenceが1つの場合には、クリップAVストリームファイルの先頭と一致し、値が「0」となる。

【0221】

フィールドnum_of_STC_sequencesは、シーケンスATC_sequence上のシーケンスSTC_sequenceの数を表す。再生専用の媒体の場合、シーケンスSTC_sequenceの数は、「1」にすることができるため、詳細は省略する。フィールドoffset_STC_idは、値が「0」に固定されるとされる。フィールドPCR_PIDは、MPEG2 TSにおいて、PCR(Program Clock Reference)が含まれるTSパケットのPIDを表す。フィールドSPN_STC_startは、シーケンスSTC_sequenceの開始をパケット番号で表したものであり、シーケンスSTC_sequenceが1つの場合には、クリップAVストリームファイルの先頭と一致し、値が「0」となる。フィールドpresentation_start_timeおよびフィールドpresentation_end_timeは、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presentation_start_timeおよびフィールドpresentation_end_time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。

【0222】

図52は、ブロックProgramInfo()の一例の構造を表すシンタクスを示す。これは、記録再生媒体用のシンタックス構造を再生専用媒体に適用することができ、新規の構造は無いので詳細な説明は省略する。ただし、フィールドnum_of_program_sequencesの値が「1」であること、フィールドnum_of_groupsの値が「1」であるなどの制限を加えることができる。

【0223】

図53は、ブロックStreamCodingInfo()一例の構造を表すシンタクスを示す。これも、上述のブロックProgramInfo()と同様、記録再生媒体用のシンタックス構造と略同じ構造

10

20

30

40

50

を有し、ビデオデータに関しては、ビデオデータのフォーマット、フレームレート、アスペクト比などの属性情報、オーディオデータに関しては、サンプリング周波数などの属性情報が記述される。ただし、再生専用媒体に適用する際には、この図 5 3 に示されるように、字幕、オーディオストリームの言語を表すフィールド language_code を追加する必要がある。この情報は、プレーヤの言語設定に従って最適な音声・字幕の言語を選択する際に有用である。

【 0 2 2 4 】

図 5 4 は、ブロック CPI() の一例の構造を表すシンタクスを示す。一般に、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定されている。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。

10

【 0 2 2 5 】

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。

【 0 2 2 6 】

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。

20

【 0 2 2 7 】

フィールド CPI_type は、CPI の種類を表し、図 5 5 に一例が示されるような値を取る。この発明の場合は、再生専用媒体用の CPI であることを示す値となる。具体的には、値が「8」とされ、HDMV 用の EP エントリのマップ (EP_map_type_for_HDMV) が格納されることが示される

【 0 2 2 8 】

図 5 6 は、再生専用媒体用の EP エントリのマップ EP_map、すなわち、上述のフィールド CPI_type 内のブロック EP_map_for_HDMV() の一例のデータ構造を表すシンタクスを示す。マップ EP_map は、GOP の先頭位置について、再生時刻とファイル内アドレスを対応付けたテーブルである。図 5 6 の例では、GOP の先頭位置について、MPEG の PTS (Presentation Time Stamp) と SPN (Source Packet Number) とが対応付けられたテーブルとしてデータベースが作られている。なお、SPN は、ソースパケット番号を示し、ファイルの先頭からのバイトアドレスに相当する。

30

【 0 2 2 9 】

記録再生用のマップ EP_map の構造と、この再生専用媒体用のマップ EP_map の構造は、略同一とされている。この実施の一形態では、データ量の削減および検索の高速化のために、各値を coarse と fine に分割し、大まかな単位での検索 (coarse) と、より精密な単位での検索 (fine) とを分けて行うことが可能なようにされている。そのため、内部構造が coarse および fine のそれぞれに対応した二つの for ループに分かれており、「GOP の最初の I ピクチャの PTS 対 ファイル内アドレス」という単純なテーブルと比べて、多少複雑になっている。

40

【 0 2 3 0 】

フィールド EP_fine_table_start_address は、上述した、より精密な検索を行うためのテーブルの位置が示される。次の for ループは、フィールド PTS_EP_coarse および SPN_EP_coarse により、大まかな単位で検索を行う (coarse) ためのテーブルが格納され、フィールド ref_to_EP_fine_id により、大まかな単位から参照される精密な検索 (fine) のためのテーブルの番号が記述される。フィールド PTS_EP_coarse および SPN_EP_coarse は、それぞれ PTS および SPN の上位側のビットが示される。

50

【 0 2 3 1 】

次に、パディングワードを挟んで、精密な検索を行うためのフィールドPTS_EP_fineおよびSPN_EP_fineが格納されるforループが配される。それと共に、フラグis_angle_change_pointおよびフィールドl_end_position_offsetが格納される。詳細は後述するが、フラグis_angle_change_pointは、クリップAVストリームがマルチアングルを構成している場合に、各EPエントリがアングル切り替え可能点に該当しているか否かを示す（第2のフラグ）。

【 0 2 3 2 】

ここで、フラグis_angle_change_pointについて説明する。まず、理解を容易とするために、クリップAVストリームがディスク上にどのように配置されているかについて説明する。

10

【 0 2 3 3 】

クリップAVストリームは、ディスク10上の論理アドレスが連続したデータのかたまり（ブロック）の並びで構成されている。以下、このクリップAVストリームによるディスク上の論理アドレスが連続したデータのブロックを、エクステントと称する。図57は、ディスク上にエクステントが配置された様子を模式的に示す。図57の例では、1つのクリップAVストリームが3つのエクステント601A、601B、601Cの並びで構成されている。エクステントの長さとその配置は、ディスク作成の際に適切に制御されている。

【 0 2 3 4 】

このとき、エクステントの最小サイズの規定を守るようにエクステント601A、601B、601Cが構成されて、クリップAVストリームがディスク600に記録されていれば、再生時にデータの連続供給が保証される。

20

【 0 2 3 5 】

図57に示されるように、1つのファイルが複数のエクステント601A、601Bおよび601Cから構成されている場合、ファイルの読み出しのために、あるエクステント601Aから次のエクステント601Bへと、次々にエクステントを読む動作が繰り返される。あるエクステント601Aを読んだ後に次のエクステント601Bを読む場合、ディスク600の回転数を変化させ、プレーヤにおいてディスク600に対する読み出し部を構成する光ピックアップを移動させる必要がある。

30

【 0 2 3 6 】

ディスク600上での、光ピックアップを移動させてのアクセス時間は、最大で、約0.8秒とされ、光ピックアップの移動中は、ディスク読み出し部を構成するドライブからは、データを取り出すことができない。一方、映像および音声のデコードおよび再生は、連続に行われるため、このアクセス時間の影響を吸収するような仕組みが必要となる。

【 0 2 3 7 】

あるエクステント601Aと次のエクステント601Bとの間では、アクセスのジャンプが発生する。前のエクステント601Aのバイト長が大きい場合、ジャンプ前に十分な量のデータをエクステント601Aから読み出し、バッファに格納することができる。このため、ジャンプする間も、バッファに格納されたデータをデコーダに供給することで、連続的にデコードを行うことができる。

40

【 0 2 3 8 】

すなわち、ジャンプ前のエクステント601Aのサイズが十分大きければ、次のエクステント601Bにジャンプしている間も、ストリームの連続的な供給が保証できる。したがって、バッファに十分な量のデータを蓄積できるようにするには、エクステントのサイズの下限を制限すればよい。最小エクステントサイズは、ディスク600からの読み出しレートと、ドライブのアクセス時間から固有的な値として定められる。データ連続供給の条件は、このエクステントの最小サイズの規定を守ってデータが配置されているか否かにより判定できる。

【 0 2 3 9 】

50

図58は、エクステント601Dおよび601Eのように、クリップAVストリームがディスク上に断片化して記録されている状態の例を示す。図58において、記録されたクリップAVストリームを、所定のビットレートで記録媒体から読み出せることを保証するためには、1つのエクステントの大きさ(サイズSとする)が、次式(1)に示されるような条件を満たす必要がある。

$$S \times 8 / (S \times 8 / R_{ud} + T_s) \geq R_{max} \cdots (1)$$

【0240】

式(1)において、「S」は、エクステントのサイズをバイトで表わし、「T_s」は、1つのエクステント601Dから次のエクステント601Eへのフルストロークのアクセス時間すなわち最大アクセス時間を秒で表わす。「R_{ud}」は、ディスクからの読み出しビットレートを表わし、「R_{max}」は、クリップAVストリームのビットレートを表わす。「R_{ud}」および「R_{max}」の単位は、ビット/秒である。なお、エクステントサイズSに乘じられている「8」は、バイト/ビットの換算のためである。

10

【0241】

すなわち、各エクステント601D、601EのサイズがSバイト以上になるように、クリップAVストリームのデータを連続して配置すれば、エクステント601Dおよび601Eの並びをシームレスに再生することが可能となる。

【0242】

式(1)を変形すると、次式(2)のようになる。

$$S \times 8 \geq R_{max} \cdot R_{ud} \cdot T_s / (R_{ud} - R_{max}) \cdots (2)$$

20

この式(2)に基づき具体的な数値を計算すると、例えば、クリップAVストリームのビットレートR_{max}=28(ビット/秒)の場合において、R_{ud}=35Mbps(Mega Bit per second)、T_s=0.8秒とすると、S≥14MB(Mega Byte)となる。つまり、この例では、最小エクステントサイズSは、14MBとなる。

【0243】

以上のようにして、一つのファイルを連続に読み出す際のデータ連続供給の条件を決めることができる。これは、マルチアングル機能において読み出すアングルをシームレスに切り替える、シームレスアングル切り替え機能の実現にも適用できる。

【0244】

図59を用いて、マルチアングル機能について説明する。マルチアングル機能は、図59Aに一例が示されるように、複数の再生経路が用意された区間(マルチアングルブロック)があり、ユーザは、その区間を再生している間、見たいアングルを自由に変えることができるという機能である。例えば、同一の対象物をアングルを変えて複数のカメラで並列的に撮影することによって、このようなマルチアングルブロックを構成することができる。

30

【0245】

なお、マルチアングルを構成する映像は、同一シーンを対象にしたものに限られない。再生時系列上の同一時刻に選択的に表示されるように意図された複数の映像により、マルチアングルを構成することができる。

【0246】

マルチアングルブロックは、1つのプレイアイテムから構成される。すなわち、1つのプレイアイテムが複数のアングルの映像を含むことになる。図59Aの例では、マルチアングルブロックにおいて、3つのアングル(アングル(0)、アングル(1)およびアングル(2))を選択可能とされている。

40

【0247】

マルチアングルブロックにおいては、アングル間を切り替える際のアクセス時間を短くするため、図59Bに一例が示されるように、マルチアングルを構成する各ストリームがディスク上に特殊な配置で記録される。これを、インタリーブブロックと呼び、一つの連続部分をインタリーブユニットと呼ぶ。インタリーブユニットは、ディスク上の連続的なアドレスに記録されており、最小エクステントサイズの条件を満たす。インタリーブプロ

50

ックは、各々のアングルの時間的に対応するインタリーブユニットをアングル順に並べて纏め、それを、さらに時系列に従って並べたものである。

【0248】

インタリーブユニットを単位にしてアングルを切り替えれば、常に最小エクステントサイズを読み込んでからジャンプをするため、上述したデータ連続供給の条件が満たされる。さらに、インタリーブユニットの先頭にGOPの先頭を合わせ、常にインタリーブユニットの先頭からMP EGデコードが可能になるようにしておけば、シームレスなアングル切り替えを実現できる。

【0249】

ここで、シームレスアングル切り替えが可能な位置について考える。フレーム間圧縮を行うMP EGストリームの性質と、ディスクからのデータ連続供給の条件から、アングル切り替えは、任意のバイト位置ではできないことがわかる。最も簡単なのは、最小エクステントのサイズを満たすインタリーブユニットの単位で切り替えられるようにする方法である。さらに、インタリーブユニットの先頭とGOPの先頭、すなわちEP エントリとを一致させれば、MP EGストリームのデコードも連続に行える。

10

【0250】

ここで、例えばアングルの切り替え可能点を増やし、インタリーブによってインタリーブブロックの数が増えると、一つのファイルが細切れになり、ファイルシステムの管理情報が増える。例えばBlu-ray Discファイルシステムでは、ファイルが複数のエクステントに分割されると、ディスクリジョンの数が増えてファイルシステムのデータベースが肥大化し、記録できるファイル数の上限が小さくなるなどの問題が出てくる。

20

【0251】

そこで、この発明の実施の一形態では、図60に一例が示されるように、一つのインタリーブユニットのサイズを大きくして、インタリーブブロック中のインタリーブユニットの数を減らし、一つのインタリーブユニット中に複数のアングル切り替え可能点を設ける。アングル切り替え可能点は、デコード可能な開始点でもあるため、何れかのEP エントリに一致する。

【0252】

上述したフラグis_angle_change_pointは、GOPの先頭を指すEP エントリがアングル切り替え可能点に該当するか否かを表すものである。フラグis_angle_change_pointは、図61に一例が示されるような値を取り、フラグis_angle_change_pointが値「1」で、当該EP エントリがアングル切り替え可能点であることを示し、値「0」で、当該EP エントリがアングル切り替え可能点ではないことが示される。

30

【0253】

あるEP エントリをシームレスなアングル切り替え点にできるかどうかは、基本的には、当該EP エントリと前のアングル切り替え可能点とが、最小エクステントのサイズ以上離れているか否かで決めることができる。すなわち、フラグis_angle_change_pointの値が「1」であるアングル切り替え可能なEP エントリの間隔は、最小エクステントサイズ以上離れている必要がある。

【0254】

次に、図60を用いて、このフラグis_angle_change_pointを利用したシームレスアングル切り替えの動作について説明する。図60において、上向きの矢印で示される点がアングル切り替え可能点である。ディスク上では、図60Bに一例が示されるように、上向きの矢印で示される点がアングル切り替え可能点とされるEP エントリであって、フラグis_angle_change_pointが値「1」とされる。

40

【0255】

図60Aを参照し、例えば、アングル再生中にユーザがアングル(0)からアングル(1)に切り替えたとする。そのとき、プレーヤは、アングル(0)の最寄りのEP エントリのうち、フラグis_angle_change_pointの値「1」になっている箇所までそのアングルのストリームを読み続ける。次に、そのEP エントリの時刻と同時刻のアングル(1)

50

中のEPエントリをクリップ情報ファイルから探し、ファイル内のバイト位置を取得し、そこから読み込みを開始する。以上の動作により、シームレスなアングル切り替えを実現できる。

【0256】

このときのディスク上での再生経路は、図60Bに一例が示されるように、フラグis_angle_change_pointが値「1」となっている位置で、アングル間でのジャンプが行われている。

【0257】

次に、ノンシームレスなアングル切り替えの場合を考える。ノンシームレスなアングル切り替えとは、アングルを切り替える際、映像に不連続が発生してもよいアングル切り替えのことをいう。シームレスアングル切り替えの場合、上述したように、アングルの切り替えが指定されると、アングル切り替え可能点まで再生してアングル切り替えが実行されるが、ノンシームレスアングル切り替えの場合は、データ連続供給やGOPの先頭の条件を考慮せずに、アングル切り替えの指定がなされて即座にアングルを切り替えることができるので、切り替えまでの時間が短いというメリットがある。

【0258】

図62は、ノンシームレスのマルチアングルブロックの例を示す。ノンシームレスのアングル切り替えの場合、画像が不連続になってもよいため、必ずしも同時刻の他アングルに移動する必要はない。例えばGOPの再生途中でアングルを抜け出し、その時刻よりも前で、最も近いGOPの先頭にジャンプしてもよい。図62Aに例示されるように、アングル(0)からアングル(1)にジャンプしたときに、アングル(1)においてGOPの先頭を探し、この例では、時間的に少し戻った位置からアングル(1)が再生されている。図62Bは、このときのディスク上での再生経路を示す。

【0259】

一方、ノンシームレスなマルチアングルブロックであっても、マルチアングルブロックの出口での接続は、シームレスであることが求められる。これは、一度もアングル切り替えを行わない場合には、通常の再生と同様にマルチアングルブロックの入口と出口で画像に不連続が発生しないようにすることが、再生品質を高める上で非常に効果的だからである。

【0260】

この発明の実施の一形態では、フラグis_angle_change_pointの設定により、ノンシームレスのマルチアングルブロックにおいて、マルチアングルブロックと本編との接続点がシームレスに接続されるようにしている。

【0261】

まず、本編からアングルブロックへの入口におけるシームレス接続は、比較的实现が容易である。すなわち、本編の終わり(すなわち、ノンシームレスのマルチアングルブロックの入り口の直前におけるプレイアイテムの後端)を、最小エクステントサイズの条件を満たすよう配置しておけば、本編からどのアングルにもシームレスに接続することができる。

【0262】

一方、アングルブロックを抜けて本編に接続する出口付近においては、例えば、図63Aに一例が示される経路のようなアングル切り替えが行われた場合、最後のアングル切り替えにおいて、切り替えを行った点からアングルの終わりまでのデータは、最小エクステントサイズを満たさない。そのため、マルチアングルブロックの出口でのジャンプにおいて、データ連続供給の条件が満たされず、映像が一時停止する、あるいは黒画面が出るなどの不連続が発生してしまう。図63Bは、このときのディスク上での再生経路を示す。

【0263】

図63Aの例では、アングル(2)からアングル(1)に移動した位置から当該マルチアングルブロックの出口までが、最小エクステントサイズを満たしていない。そのため、アングル(1)から本編に移動する際に、アングル(1)のデータ読み込み量が不足し、

10

20

30

40

50

再生映像の不連続が発生する。

【0264】

この発明の実施の一形態では、このノンシームレスのアングル切り替えにおける問題を解決するために、ノンシームレスアングル切り替えのブロックにおいても、フラグis_angle_change_pointを活用する。ノンシームレスのマルチアングルブロックの再生中においては、アングル切り替え時に不連続が発生しても良いので、基本的にフラグis_angle_change_pointの値は、上述の図63Bに示されるように、全て「1」である。すなわち、任意の箇所再生中のアングルから飛び出すことが可能であると共に、任意の箇所に飛び込んで再生をすることが許される。

【0265】

ここで、ノンシームレスのマルチアングルブロック中のクリップにおける、フラグis_angle_change_pointの定義を定める。上述したように、フラグis_angle_change_pointの値が「0」の領域は、再生中における他のアングルへのアングル切り替え（飛び出し）や、切り替えが発生した元のアングルからの再生位置の割り込み（飛び込み）が禁止され、フラグis_angle_change_pointの値が「1」の領域は、飛び出しや飛び込みが可能であるものとする。そして、各アングルの出口付近の所定数のフラグis_angle_change_pointの値を「0」に設定し、この所定数のフラグis_angle_change_pointの値が「0」にされた領域での飛び出しや飛び込みを禁止する。このとき、この所定数のフラグis_angle_change_pointの値が「0」とされた領域のサイズは、最小エクステントサイズ以上とする。

【0266】

このようにフラグis_angle_change_pointを設定することで、ノンシームレスのマルチアングルブロックにおいても、マルチアングルブロックの出口の直前での飛び込み、飛び出しを禁止することができる。

【0267】

例として、図64Bおよび図65Bに示されるように、各インタリーブユニットの後端側の6個のEPエントリをフラグis_angle_change_pointの値が「0」である領域とし、他のEPエントリは、フラグis_angle_change_pointの値が「1」であるように設定する。

【0268】

次に、このようにフラグis_angle_change_pointが設定されたノンシームレスのマルチアングルブロックの再生方法について説明する。第1の方法は、フラグis_angle_change_pointの値が「0」の領域でのアングル切り替えを禁止する方法である。

【0269】

図64を用いて説明する。上述したように、各インタリーブユニットの後端側の6個のEPエントリにおいて、フラグis_angle_change_pointの値を「0」とする（図64B）。図64Aに一例が示されるように、再生位置がこのフラグis_angle_change_pointの値を「0」とした領域に入ったら、アングル切り替えが禁止される。例えば、ユーザによりこの領域でアングル切り替えを指示する操作が行われても、プレーヤでは、無視される。この例では、アングル切り替えが行われず、そのままマルチアングルブロックを抜けて本編のプレイアイテムに移行している。この場合のディスク上での再生経路が図64Bに示される。

【0270】

第2の方法は、フラグis_angle_change_pointの値が「0」の領域でのアングル切り替えは許可されるが、切り替え後の飛び込み先は、フラグis_angle_change_pointの値が「1」の箇所とする方法である。

【0271】

図65を用いて説明する。上述したように、各インタリーブユニットの後端側の6EPエントリにおいて、フラグis_angle_change_pointの値を「0」とする（図65B）。図65Aに一例が示されるように、マルチアングルブロックにおいて、ブロックの出口から最小エクステントサイズ未満の位置、すなわち、フラグis_angle_change_pointの値が「

10

20

30

40

50

0」とされた領域内でアングル切り替えが指定された場合、切り替えられた先のアングルにおいて、再生位置がフラグis_angle_change_pointの値が「1」の領域まで戻され、その位置から再生が継続される。この場合のディスク上での再生経路が図65Bに示される。

【0272】

これら第1および第2の方法の何れを用いても、マルチアングルブロックの出口でのジャンプの際には、最小エクステントサイズ以上のデータを読み込んだ後に次のプレイアイテムの再生が行われる。これにより、アングルブロックの出口で再生が不連続（ノンシームレス）になることを防止できる。

【0273】

なお、上述の図64では、ノンシームレスのマルチアングルブロックにおいてインタリーブブロックを構成したが、ノンシームレスのマルチアングルブロックにおいては、図65に示されるように、インタリーブ配置にしなくてもブロックを構成できる。

【0274】

図66は、この発明の実施の一形態に適用できるプレーヤデコーダ100'の一例の構成を示す機能ブロック図である。このプレーヤデコーダ100'は、図示されないドライブ装置に装填されたディスクから再生されたデータを解釈し、AVストリームを出力すると共に、出力されたAVストリームに対するユーザによるインタラクティブな操作を可能とする。

【0275】

なお、プレーヤデコーダ100'は、図示されないCPUにより全体の動作が制御される。例えば、プレーヤデコーダ100'の各部におけるストリームやデータの流れは、CPUにより監視され、制御される。

【0276】

図示されないドライブ装置にディスクが装填されると、上述したように、先ずファイル「scenario.hdmv」およびファイル「entrylist.data」が再生され、このファイル「scenario.hdmv」およびファイル「entrylist.data」の記述に基づき、必要な他のファイルが読み出され、ディスクに記録されたコンテンツが再生される。例えば、ファイル「scenario.hdmv」およびファイル「entrylist.data」の記述に基づき、動画プレーン10に表示するための動画データ、字幕プレーン11やグラフィクスプレーン12に表示するための画像データ、プレイリストファイルなどがディスクから読み出される。

【0277】

以下では、ディスクから読み出されるこれらのデータのうち、動画データ、サブピクチャ（字幕データ）や音声データといった、連続的に処理する必要があるストリームをリアルタイムストリームと称する。また、シナリオファイル、プレイリストファイルといった、連続的な処理を要求されない非リアルタイムなデータを、ストアオブジェクトと称する。ストアオブジェクトは、メモリ上などに蓄積、展開され、必要に応じて処理される。

【0278】

プレーヤデコーダ100'は、チャンネル(1)および(2)の2系統の入力チャンネルを有し、入力チャンネル(1)の入力端101に、ストアオブジェクトが入力される。入力チャンネル(2)の入力端202に、リアルタイムストリームが入力される。入力端202に、ストアオブジェクトを入力することも可能である。この実施の一形態では、入力端202に入力されるリアルタイムストリームおよび一部のストアオブジェクトは、MPEG2 TSである。

【0279】

なお、例えば、ドライブ装置においてディスクの回転速度を2倍速などの高速回転としてディスクからの読み出し転送レートを上げ、時分割で動作させることにより、ディスクからの、チャンネル(1)および(2)の2系統の読み出しが実現可能である。

【0280】

先ず、入力チャンネル(1)の系統について説明する。入力端101に入力されたスト

10

20

30

40

50

アオブジェクトは、スイッチ回路 102 に入力される。ストアオブジェクトとして、ファイル「scenario.hdmv」中のプログラムコードが入力された場合、スイッチ回路 102 において出力端 102A が選択され、入力されたプログラムコードがコードバッファ 104 に蓄えられる。他にプログラムとしては、HTML 文書や、ECMA スクリプトなども考えられる。

【0281】

一方、ストアオブジェクトとして画像データや音声データが入力された場合、スイッチ回路 102 において出力端 102B が選択され、入力された画像データがスイッチ回路 103 に入力される。入力端 202 に入力されたリアルタイムストリームに、字幕プレーン 11 やグラフィクスプレーン 12 に表示するための画像データが含まれていない場合には、スイッチ回路 103 で入力端 103A が選択され、スイッチ回路 102 から入力された画像データや音声データがコンテンツバッファ 105 に蓄えられる。

10

【0282】

同様にして、入力端 202 に入力されたリアルタイムストリームに、字幕プレーン 11 やグラフィクスプレーン 12 に表示するための画像データや、効果音のための音声データが含まれている場合には、スイッチ回路 103 において入力端 103B が選択され、当該画像データ・音声データがコンテンツバッファ 105 に蓄えられる。コードバッファ 104 およびコンテンツバッファ 105 に蓄えられたストアオブジェクトは、必要に応じて読み出され、マルチメディアエンジン 106 に供給される。

20

【0283】

コンテンツバッファ 105 に蓄えられたストアオブジェクトのうち画像データは、スイッチ回路 107 および 108 をそれぞれ介して、グラフィックデコーダ A 500 およびグラフィックデコーダ B 301 にも供給される。

【0284】

なお、この図 66 の例では、グラフィクスデコーダ A 300 は、PNG 形式の画像データをデコードし、グラフィクスデコーダ B 301 は、JPEG 形式の画像データをデコードする。なお、グラフィクスデコーダ A 300 および B 301 は、これに限らず、さらに他のデータ形式の画像データをデコードするものであってもよいし、複数の形式の画像データに対応するようにもできる。

30

【0285】

マルチメディアエンジン 106 は、XML パーサ 106A、スクリプトインタプリタ 106B およびグラフィクスレンダラ 106C、効果音再生のためのサウンドプレーヤ 106D、シナリオファイル解析エンジン 106E を含む。マルチメディアエンジン 106 は、独立的なハードウェアで構成してもよいし、上述した図示されない CPU の、所定のプログラムに基づく処理で実現することも可能である。

【0286】

XML パーサ 106A は、XML (Extensible Markup Language) 文書を解析する機能を有し、HTML 文書の解析も可能である。XML パーサ 106A で解釈された HTML 文書は、このプレーヤデコーダ 100 で実行可能な形式に変換される。スクリプトインタプリタ 106B は、ECMA スクリプトを解析し、このプレーヤデコーダ 100 で実行可能な形式に変換される。また、グラフィクスレンダラ 106C は、画像データを、字幕プレーン 11 およびグラフィクスプレーン 12 に展開可能な形式にデコードする。サウンドプレーヤ 106D は、ボタンクリック音などに使われる音声データを再生する。

40

【0287】

マルチメディアエンジン 106 において、バッファ 109 をワークメモリとして、これら XML パーサ 106A、スクリプトインタプリタ 106B およびグラフィクスレンダラ 106C の処理が行われる。例えば、XML パーサ 106A およびスクリプトインタプリタ 106B により、バッファ 109 のうちコードバッファ 109A が用いられる。また、グラフィクスレンダラ 106C により、バッファ 109 のうちグラフィクスバッファ 109D が用いられる。バッファ 109 は、上述のコードバッファ 109A およびグラフィク

50

スバッファ109Dの他に、文字列の表示に用いるフォントデータが格納されるフォントバッファ109B、XMLパーサ106AでHTML文書を解析した結果を階層化された木構造で保持するためのツリーバッファ109Cなどが含まれる。サウンドプレーヤ106Dは、サウンドバッファ109Eに蓄積した音声データを再生する。

【0288】

マルチメディアエンジン106では、例えばコードバッファ104に蓄えられたシナリオファイル「scenario.hdmv」などを読み出し、読み出されたファイルの記述に基づきシナリオ解析エンジン106Eでシナリオを解析する。そして、解析されたシナリオの記述に基づき図示されないドライブ装置などを制御し、ドライブ装置に挿入されたディスクから、動画プレーン10に表示するための動画データ、字幕プレーン11やグラフィクスプレーン12に表示するための画像データ、音声データなど必要なファイルを読み出し、コンテンツを再生する。

10

【0289】

なお、コードバッファ104およびコンテンツバッファ105に格納されたデータは、当該データが不要になるまで、コードバッファ104やコンテンツバッファ105に保持しておくことができる。したがって、これらコードバッファ104やコンテンツバッファ105に格納されたデータは、必要に応じて何度でも読み出して使うことができる。

【0290】

さらに、マルチメディアエンジン106では、例えば、コードバッファ104に蓄えられたECMASクリプトを読み出し、読み出されたECMASクリプトの記述に基づき、必要に応じて、コードバッファ104からの他のECMASクリプトやHTML文書の読み出し、コンテンツバッファ105からの画像データの読み出しなどを行う。

20

【0291】

マルチメディアエンジン106では、上述の他にも、入力された複数種類のデータのマルチプレクス処理、JavaVM（Java（登録商標）仮想マシン）機能などが行われる。さらに、マルチメディアエンジン106により、ユーザからの、リモートコントロールコマンドやポインティングデバイスなどによる図示されない操作手段からの入力を受け取られ、所定に処理される。ユーザ入力は、さらに、後述するグラフィックデコーダA300、グラフィックデコーダB301、オーディオデコーダ118、MPEGビデオデコーダ120およびシステムデコーダ121にも供給される。

30

【0292】

グラフィクスレンダラ106Cで処理された画像データは、スイッチ回路130および131をそれぞれ介してサブピクチャプレーン302およびグラフィクスプレーン303に供給される。なお、この例では、サブピクチャプレーン502およびグラフィクスプレーン303に供給される画像データは、PNG形式あるいはJPEG形式等の画像データをグラフィクスレンダラ106Cでデコードした後のビットマップデータであるものとする。これらの各プレーン302、303に画像データが供給されるタイミングは、マルチメディアエンジン106により制御される。

【0293】

ここで、サブピクチャプレーン302およびグラフィクスプレーン303は、それぞれ上述した字幕プレーン11およびグラフィクスプレーン12に対応する。動画像プレーン135は、上述した動画プレーン10に対応する。なお、サブピクチャプレーン302、グラフィクスプレーン303および動画像プレーン135は、例えばフレームメモリからなる。

40

【0294】

マルチメディアエンジン106は、さらに、後述するプレゼンテーションプロセッサ139に対して、動画像プレーン135、サブピクチャプレーン302、ならびに、グラフィクスプレーン303を切り替えあるいは合成する制御信号を供給する。同様に、マルチメディアエンジン106は、後述するプレゼンテーションプロセッサ141に対して、オーディオストリーム出力を制御するような制御信号を供給する。

50

【0295】

次に、入力チャンネル(2)の系統について説明する。入力端202にMPEG2 TSで入力されたリアルタイムストリームは、PIDフィルタ110に供給され、MPEG2 TSのトランスポートパケットに格納されるPID(Packet Identification)が抽出され、当該トランスポートパケットに格納されるストリームの属性が検出される。PIDフィルタ110では、このストリーム属性に基づき、入力されたリアルタイムストリームが、トランスポートパケット毎に対応する系統に振り分けられる。

【0296】

PIDに基づき、トランスポートパケットがストアオブジェクトに属する画像データが格納されているパケットであるとされれば、当該トランスポートパケットは、バッファTBn111Aに一旦溜め込まれ、所定のタイミングで読み出されて入力端103Bが選択されたスイッチ回路103に入力され、スイッチ回路103を介してコンテンツバッファ105に格納される。

10

【0297】

PIDフィルタ110において、PIDに基づき、トランスポートパケットがPNG形式のデータが格納されているパケットであるとされれば、当該トランスポートパケットは、バッファTBn111BおよびバッファBn112Bに一旦溜め込まれ、所定のタイミングで読み出されて入力端107Bが選択されたスイッチ回路107に入力され、スイッチ回路107を介してグラフィックデコーダA500に供給される。

【0298】

グラフィックデコーダA300では、供給されたトランスポートパケットのヘッダ情報を除去すると共に、当該トランスポートパケットに格納されたPNGデータがデコードされて字幕あるいはグラフィックスを表示するための画像データとされる。この画像データを字幕として用いる場合は、所定のタイミングでスイッチ回路130の入力端130Bに入力され、スイッチ回路130を介してサブピクチャプレーン302に展開される。同様に、ボタンとして用いる場合は、グラフィックスプレーン上に表示させるため、所定のタイミングでスイッチ回路131の入力端131Cに入力され、スイッチ回路131を介してグラフィックスプレーン303に展開される。

20

【0299】

PIDフィルタ110において、PIDに基づき、トランスポートパケットが、JPEGデータが格納されているパケットであるとされれば、当該トランスポートパケットは、バッファTBn111CおよびバッファBn112Cに一旦溜め込まれ、所定のタイミングで読み出されて入力端108Bが選択されたスイッチ回路108に入力され、スイッチ回路108を介してグラフィックデコーダB500に供給される。

30

【0300】

グラフィックデコーダB301では、供給されたトランスポートパケットのヘッダ情報を除去すると共に、当該トランスポートパケットに格納されたJPEGデータがデコードされ、ビットマップの画像データとされる。この画像データを字幕として用いる場合は、所定のタイミングでスイッチ回路130の入力端130Cに入力され、スイッチ回路130を介してサブピクチャプレーン302に展開される。同様に、ボタンとして用いる場合は、グラフィックスプレーン上に表示させるため、所定のタイミングでスイッチ回路131の入力端131Bに入力され、スイッチ回路131を介してグラフィックスプレーン302に展開される。

40

【0301】

PIDフィルタ110において、PIDに基づき、トランスポートパケットが、オーディオデータが格納されているパケットであるとされれば、当該トランスポートパケットは、バッファTBn111DおよびバッファBn112Dに一旦溜め込まれ、所定のタイミングで読み出されてオーディオデコーダ118に供給される。このトランスポートパケットに格納されるオーディオデータは、例えばドルビーデジタルに準拠した方式で圧縮符号化されている。

50

【 0 3 0 2 】

オーディオデコーダ 1 1 8 は、例えばリニア P C M (Pulse Code Modulation) オーディオデコーダ 1 1 9 を有し、入力されたトランスポートストリームのヘッダ情報を除去すると共に、当該トランスポートパケットに格納された圧縮符号化されたオーディオデータをデコードし、最終的にリニア P C M オーディオデータに変換する。当該トランスポートパケットに、圧縮符号化されていないリニア P C M オーディオデータが格納されている場合は、当該データは、例えばそのままリニア P C M オーディオデコーダ 1 1 9 に入力され、何も処理されずに出力される。

【 0 3 0 3 】

オーディオデコーダ 1 1 8 から出力されたリニア P C M オーディオデータは、オーディオ用のプレゼンテーションプロセッサ 1 4 1 に入力され、マルチメディアエンジン 1 0 6 の制御に基づき所定の音響効果などが付加されて、出力端 1 4 2 に導出される。

10

【 0 3 0 4 】

P I D フィルタ 1 1 0 において、P I D に基づき、トランスポートパケットが動画データが格納されているパケットであるとされれば、当該トランスポートパケットは、バッファ T B n 1 1 1 E、バッファ M B n 1 1 3 およびバッファ E B n 1 1 4 に一旦溜め込まれ、所定のタイミングで読み出されて M P E G ビデオデコーダ 1 2 0 に供給される。このトランスポートパケットに格納される動画データは、M P E G 2 方式により圧縮符号化されている。

【 0 3 0 5 】

M P E G ビデオデコーダ 1 2 0 では、供給されたトランスポートパケットのヘッダ情報を除去すると共に、当該トランスポートパケットに格納された、M P E G 2 方式で圧縮符号化された動画データをベースバンドの動画データにデコードする。

20

【 0 3 0 6 】

M P E G デコーダ 1 2 0 から出力された動画データは、スイッチ回路 1 2 4 の入力端 1 2 4 A に入力されると共に、バッファ 1 2 3 を介してスイッチ回路 1 2 4 の入力端 1 2 4 B に入力される。スイッチ回路 1 2 4 において、所定のタイミングで入力端 1 2 4 A および 1 2 4 B が選択され、出力された動画データが動画プレーン 1 3 4 に展開される。

【 0 3 0 7 】

P I D フィルタ 1 1 0 において、P I D に基づき、トランスポートパケットがシステム情報が格納されているパケットであるとされれば、当該トランスポートパケットは、バッファ T B n 1 1 1 F および B s y s 1 1 5 を介してシステムデコーダ 1 2 1 に供給される。システムデコーダ 1 2 1 では、供給されたトランスポートパケットのヘッダ情報が除去され、格納されているシステム情報が取り出される。システム情報は、例えば図示されない C P U に渡される。

30

【 0 3 0 8 】

サブピクチャプレーン 3 0 2 上の画像データは、上述の図 1 7 におけるパレットテーブル 2 2 に対応するパレット 3 0 4 に供給され、2 5 6 色からなるパレットに対してインデックスによる参照がなされ、R G B データが出力されると共に、不透明度データ 1 が抜き出される。R G B データは、上述の図 1 7 における R G B / Y C b C r 変換回路 2 9 に対応する R G B / Y C b C r 変換回路 3 0 6 により Y C b C r データに変換される。Y C b C r データは、不透明度データ 1 と共にプレゼンテーションプロセッサ 1 3 9 に供給される。

40

【 0 3 0 9 】

グラフィクスプレーン 3 0 3 上の画像データは、上述の図 1 7 におけるパレットテーブル 2 6 A に対応するパレット 3 0 5 に供給され、R G B データと不透明度データ 2 とが出力される。R G B データは、上述の図 1 7 における R G B / Y C b C r 変換回路 2 6 B に対応する R G B / Y C b C r 変換回路 3 0 7 に供給され、カラーシステムが R G B (4 : 4 : 4) から Y C b C r (4 : 4 : 4) に変換される。R G B / Y C b C r 変換回路 3 0

50

7から出力されたYCbCrデータは、不透明データ 2と共に、プレゼンテーションプロセッサ139に供給される。

【0310】

動画像プレーン135の出力は、アップ/ダウンコンバータ138を介してプレゼンテーションプロセッサ139に供給される。なお、アップ/ダウンコンバータ138は、画像の解像度を変換する回路であって、例えば高解像度のHD(High Definition)画像から通常の解像度を有するSD(Standard Definition)画像への変換を行う。

【0311】

プレゼンテーションプロセッサ139は、図17を用いて説明した、字幕プレーン11(サブピクチャプレーン302)の画像データによる不透明度 1と、グラフィクスプレーン12(グラフィクスプレーン303)による不透明度 2とを用いたアルファブレンディング処理を行う。

10

【0312】

すなわち、プレゼンテーションプロセッサ139では、サブピクチャプレーン302の画像データに設定された不透明度 1に基づき、動画像プレーン135に対してサブピクチャプレーン302の画像データが合成される。さらに、グラフィクスプレーン303の画像データに設定された不透明度 2に基づき、動画像プレーン135およびサブピクチャプレーン302が合成された画像データに対してグラフィクスプレーン303の画像データが合成される。この、グラフィクスプレーン303の画像データ、サブピクチャプレーン302の画像データ(字幕データ)および動画像プレーン135の画像データが合成された画像データが出力端140に導出される。

20

【0313】

なお、プレゼンテーションプロセッサ139は、画像データに対してリアルタイムでエフェクト処理を行うこともできる。

【0314】

このような構成において、この発明の実施の一形態に基づき記述およびデータ構成されたコンテンツが記録されたディスクが図示されないドライブ装置に挿入されると、ファイル「scenario.hdmv」などが読み込まれ、ファイル「scenario.hdmv」の記述に従い、ディスクに記録されたコンテンツが再生される。ユーザにより、図示されない操作手段により指示がなされると、この指示に基づき、マルチメディアエンジン106によりドライブ装置やプレーヤデコーダ100'の各部が制御され、指示に応じた再生がなされる。

30

【0315】

例えば、図示されない操作手段によりアングル切り替え指示がなされると、この指示に基づくマルチメディアエンジン106の制御により、ファイル「scenario.hdmv」、「entrylist.data」など必要なファイルが参照され、クリップ切り替え可能点の情報に基づき図示されないドライブ装置が制御され、上述したようにして、ディスク上の読み出し位置が変更されてアングル切り替えがなされる。また、操作手段により、静止画表示の際のポーズ設定の解除などの指示を出すこともできる。

【0316】

上述では、プレーヤデコーダ100'の各部がハードウェアで構成されるように説明したが、これはこの例に限られない。例えば、プレーヤデコーダ100'をソフトウェア上の処理として実現することも可能である。この場合、プレーヤデコーダ100'をコンピュータ装置上で動作させることができる。また、プレーヤデコーダ100'をハードウェアおよびソフトウェアが混合された構成で実現することもできる。例えば、オーディオデコーダ118やMPEGビデオデコーダ120をハードウェアで構成し、その他をソフトウェアで構成することが考えられる。

40

【0317】

プレーヤデコーダ100'をソフトウェアのみ、または、ハードウェアおよびソフトウェアの混合により構成し、コンピュータ装置で実行させるためのプログラムは、例えばCD-ROM(Compact Disc-Read Only Memory)といった記録媒体に記録されて提供される

50

。このCD-ROMをコンピュータ装置のCD-ROMドライブに装填し、CD-ROMに記録されたプログラムを所定にコンピュータ装置にインストールすることで、上述の処理をコンピュータ装置上で実行可能な状態とすることができる。なお、コンピュータ装置の構成は、極めて周知であるため、説明は省略する。

【0318】

以上説明したように、この発明は、記録済みの大容量ディスクにおいて、記録再生用のBlu-ray Disc規格を拡張することで、ユーザの入力に応じて再生が変化するインタラクティブ機能を実現することができるようになる。

【0319】

また、この発明の実施の一形態によれば、タイトルとトップメニューを構成するプレイリストの情報をデータベースに記述してあるため、画面上の選択によらず、リモコンキーなどで直接指定のタイトルを再生するしたり、トップメニュー画面を呼び出すことができるようになる。

10

【0320】

また、この発明の実施の一形態によれば、デコード単位の先頭毎にアングル切り替え可能か否かを示すフラグを設けたので、アングル切り替え点で不連続が発生しないシームレスなアングル切り替え機能を実現することができる。

【0321】

さらに、この発明の実施の一形態によれば、サブプレイアイテムがメインのプレイアイテムと非同期で再生可能か否かを示すフラグを設けたので、メインパスに対して非同期に繰り返し再生されるBGMを実現することができる。

20

【0322】

さらにまた、この発明の実施の一形態によれば、静止画とそれに付随する字幕やグラフィクスとをカプセル化して記録しているため、静止画を切り替えながら再生する、スライドショーなどを行う場合のアクセス性が向上する。

【符号の説明】

【0323】

- 10 動画プレーン
- 11 字幕プレーン
- 12 グラフィクスプレーン
- 22 パレット
- 30 BD仮想プレーヤ
- 31 プレーヤコマンド
- 32 共通パラメータ
- 40 プレイバックコントロールプログラム
- 41 メソッド
- 60 メニュー画面
- 70 シナリオ
- 73A ~ 73M プレイリスト
- 100 プレーヤデコーダ
- 104 コードバッファ
- 105 コンテンツバッファ
- 106 マルチメディアエンジン
- 109 バッファ
- 110 PIDフィルタ
- 116 サブピクチャデコーダ
- 117 静止画像デコーダ
- 118 オーディオデコーダ
- 120 MPEGビデオデコーダ
- 132 サブピクチャプレーン

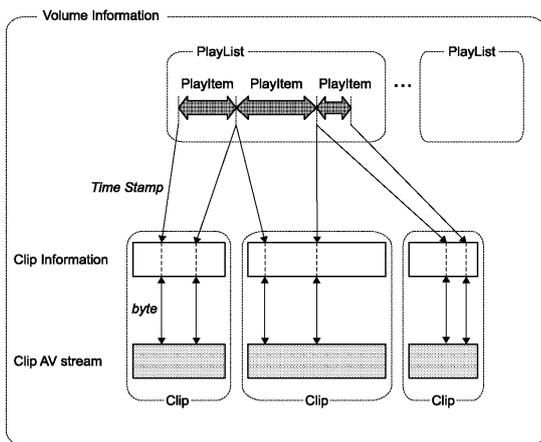
30

40

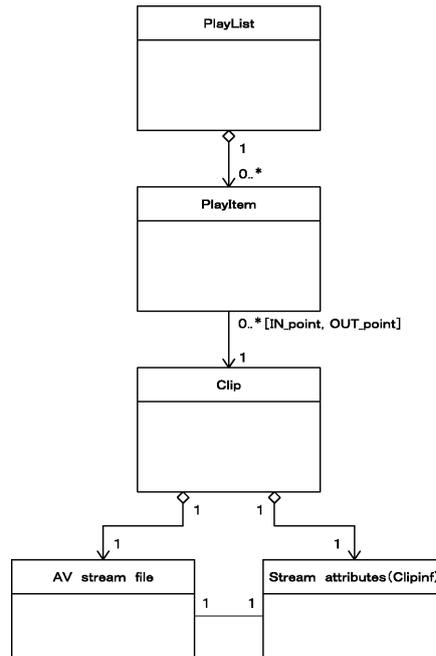
50

- 1 3 3 グラフィクスプレーン
- 1 3 4 静止画像プレーン
- 1 3 5 動画像プレーン
- 2 2 6 PNGデコーダバッファ
- 2 2 7 PNGデコーダ
- 2 2 8 オブジェクトバッファ
- 2 2 9 プレーンバッファ
- 2 3 1 オーディオミキサ
- 3 0 0 グラフィクスデコーダ A
- 3 0 1 グラフィクスデコーダ B
- 3 0 2 サブピクチャプレーン
- 3 0 3 グラフィクスプレーン
- 6 0 1 A , 6 0 1 B , 6 0 1 C , 6 0 1 D , 6 0 1 E エクステン

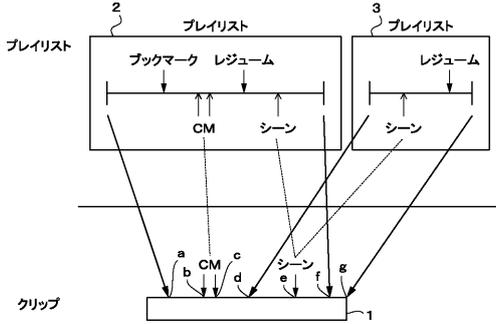
【 図 1 】



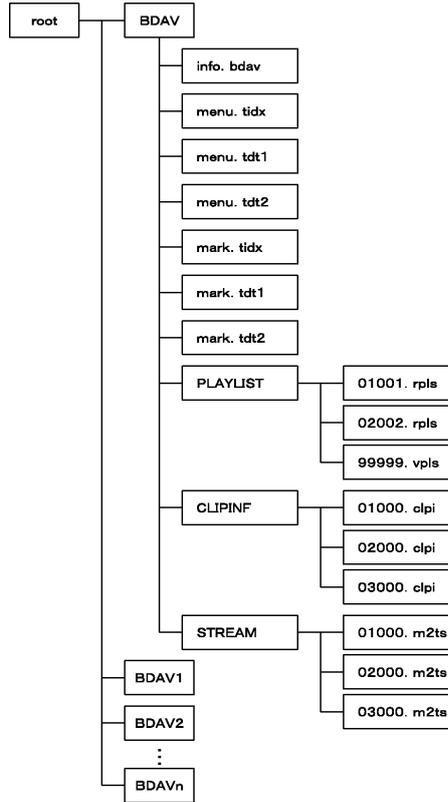
【 図 2 】



【 図 3 】



【 図 4 】



【 図 5 】

シンタクス	データ長 (ビット)	Mnemonics
info.bdav{		
type_indicator	8*4	bslbf
version_number	8*4	bslbf
TableOfPlayLists_start_address	32	unimsbf
MakersPrivateData_start_address	32	unimsbf
reserved_for_future_use	192	bslbf
UIAppInfoBDAV()		
for(i=0; i<N1; i++){		
padding_word	16	bslbf
}		
TableOfPlayLists()		
for(i=0; i<N2; i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
for(i=0; i<N3; i++){		
padding_word	16	bslbf
}		
}		

【 図 6 】

シンタクス	データ長 (ビット)	Mnemonics
UIAppInfoBDAV(){		
length	32	unimsbf
reserved_for_future_use	16	bslbf
BDAV_character_set	8	bslbf
reserved_for_word_align	6	bslbf
BDAV_protect_flag	1	bslbf
resume_valid_flag	1	bslbf
PIN	8*4	bslbf
resume_PlayList_file_name	8*10	bslbf
ref_to_menu_thumbnail_index	16	unimsbf
BDAV_name_length	8	unimsbf
BDAV_name	8*255	bslbf
}		

【 図 7 】

シンタクス	データ長 (ビット)	Mnemonics
TableOfPlayLists(){		
length	32	unimsbf
number_of_PlayLists	16	unimsbf
for(i=0; i<number_of_PlayLists; i++){		
PlayList_file_name	8*10	bslbf
}		
}		

【 図 8 】

シンタクス	データ長 (ビット)	Mnemonics
xxxx.rpls/yyyy.vpls{		
type_indicator	8*4	bslbf
version_number	8*4	bslbf
Playlist_start_address	32	unimsbf
PlaylistMark_start_address	32	unimsbf
MakersPrivateData_start_address	32	unimsbf
reserved_for_future_use	160	bslbf
UIAppInfoPlaylist()		
for(i=0;i<N1;i++){		
padding_word	16	bslbf
}		
Playlist()		
for(i=0;i<N2;i++){		
padding_word	16	bslbf
}		
PlaylistMark()		
for(i=0;i<N3;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
for(i=0;i<N4;i++){		
padding_word	16	bslbf
}		
}		

【 図 9 】

シンタクス	データ長 (ビット)	Mnemonics
UIAppInfoPlaylist(){		
length	32	unimsbf
reserved_for_future_use	16	bslbf
Playlist_character_set	8	unimsbf
reserved_for_word_align	4	bslbf
playback_protect_flag	1	bslbf
write_protect_flag	1	bslbf
is_played_flag	1	bslbf
is_edited_flag	1	bslbf
time_zone	8	bslbf
reserved_for_word_align	8	bslbf
record_time_and_date	4*14	bslbf
Playlist_duration	4*6	bslbf
maker_ID	16	unimsbf
maker_model_code	16	unimsbf
channel_number	16	unimsbf
reserved_for_word_align	8	bslbf
channel_name_length	8	unimsbf
channel_name	8*20	bslbf
Playlist_name_length	8	unimsbf
Playlist_name	8*255	bslbf
Playlist_detail_length	16	unimsbf
Playlist_detail	8*1200	bslbf
}		

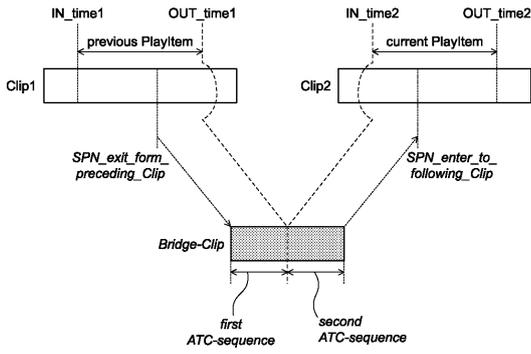
【 図 1 0 】

シンタクス	データ長 (ビット)	Mnemonics
Playlist(){		
length	32	unimsbf
reserved_for_word_align	12	bslbf
PL_CPI_type	4	bslbf
number_of_PlayItems	16	unimsbf
if(<Virtual-Playlist>&&PL_CPI_type==1){		
number_of_SubPlayItems	16	unimsbf
}else{		
reserved_for_word_align	16	bslbf
}		
for(Playlist_id=0;		
Playlist_id<number_of_PlayItems;		
Playlist_id++){		
PlaylistItem()		
}		
if(<Virtual-Playlist>&&CPI_type==1){		
for(i=0;i<number_of_SubPlayItems;i++){		
SubPlayItem()		
}		
}		
}		

【 図 1 1 】

シンタクス	データ長 (ビット)	Mnemonics
PlayItem(){		
length	16	unimsbf
Clip_Infomation_file_name	8*5	bslbf
Clip_codec_identifier	8*4	bslbf
reserved_for_future_use	6	bslbf
connection_condition	2	bslbf
if(CPI_type==1){		
ref_to_STC_Id	8	unimsbf
}else{		
reserved_for_word_align	8	bslbf
}		
IN_time	32	unimsbf
OUT_time	32	unimsbf
if(<Virtual-Playlist>		
connection_condition==3){		
&&		
BridgeSequenceInfo()		
}		
}		

【 図 1 2 】



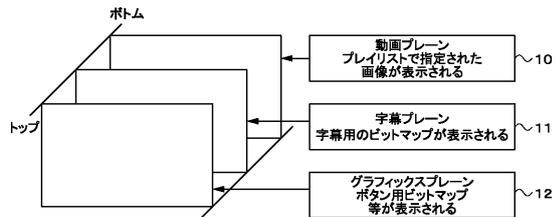
【 図 1 3 】

シンタクス	データ長 (ビット)	Mnemonics
PlaylistMark(){		
length	32	unimsbf
number_of_PlayList_marks	16	unimsbf
for(i=0;i<number_of_PlayList_marks;i++){		
mark_invalid_flag	1	unimsbf
mark_type	7	unimsbf
mark_name_length	8	unimsbf
maker_ID	16	unimsbf
ref_to_PlayItem_id	16	unimsbf
mark_time_stamp	32	unimsbf
entry_ES_PID	16	unimsbf
if(mark_type==0x01 mark_type==0x02){		
ref_to_menu_thumbnail_index	16	unimsbf
}else{		
ref_to_menu_thumbnail_index	16	unimsbf
}		
duration	32	unimsbf
makers_infomation	32	bslbf
mark_name	8*24	bslbf
}		
}		

【 図 1 4 】

シンタクス	データ長 (ビット)	Mnemonics
zzzzz.cip(){		
type_indicator	8*4	bslbf
version_number	8*4	bslbf
SequenceInfo_start_address	32	unimsbf
ProgramInfo_start_address	32	unimsbf
CPI_start_address	32	unimsbf
ClipMark_start_address	32	unimsbf
MakersPrivateData_start_address	32	unimsbf
reserved_for_future_use	96	bslbf
ClipInfo(){		
for(i=0;i<N1;i++){		
padding_word	16	bslbf
}		
SequenceInfo(){		
for(i=0;i<N2;i++){		
padding_word	16	bslbf
}		
ProgramInfo(){		
for(i=0;i<N3;i++){		
padding_word	16	bslbf
}		
CPI(){		
for(i=0;i<N4;i++){		
padding_word	16	bslbf
}		
ClipMark(){		
for(i=0;i<N5;i++){		
padding_word	16	bslbf
}		
MakersPrivateData(){		
for(i=0;i<N6;i++){		
padding_word	16	bslbf
}		
}		
}		

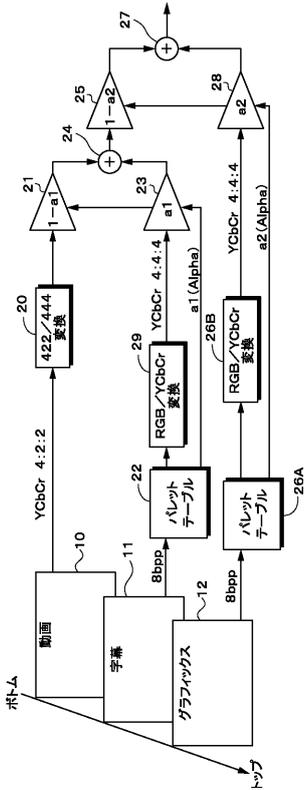
【 図 1 5 】



【 図 1 6 】

項目	規定内容
動画プレーン	1920x1080x16ビット,YCbCr(4:2:2),各8ビット
字幕プレーン	1920x1080x8ビット,8ビットカラーマップアドレス(パレット)+256段階のアルファブレンディング
グラフィックスプレーン	1920x1080x8ビット,8ビットカラーマップアドレス(パレット)+256段階のアルファブレンディング

【図 17】



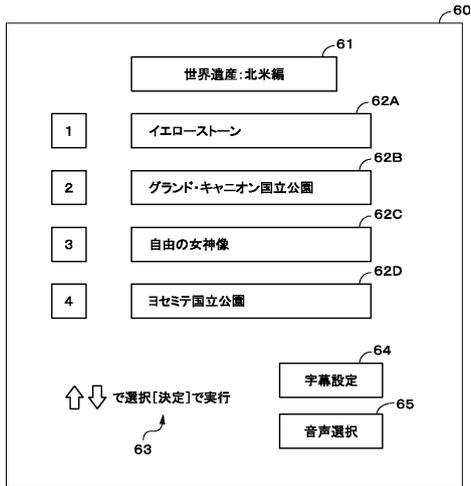
【図 18】

入力	入力アドレス 8ビット
出力	出力データ 8ビット×4、(R, G, B, α)出力

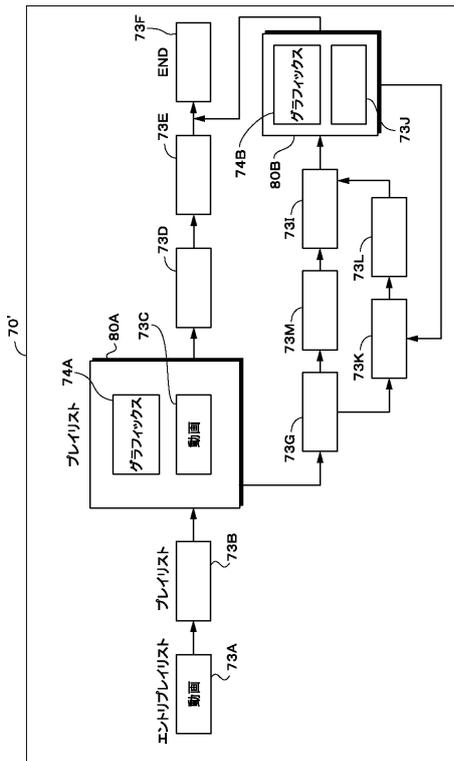
【図 19】

カラーインデックス値	3原色の値			不透明度
	R	G	B	α
0x00	0	0	0	0
0x01	10	100	30	0.5
⋮	⋮	⋮	⋮	
⋮	⋮	⋮	⋮	
0xFF	200	255	100	0.8

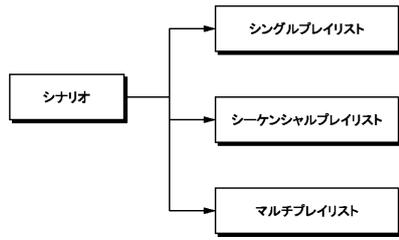
【図 20】



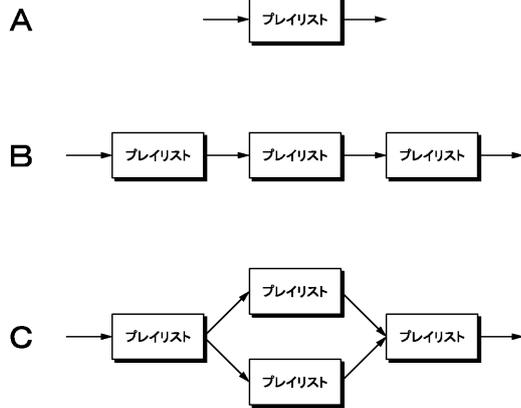
【図 21】



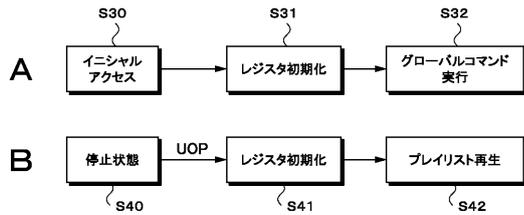
【 図 2 2 】



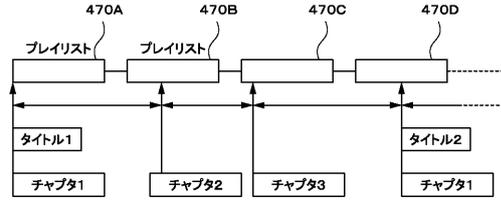
【 図 2 3 】



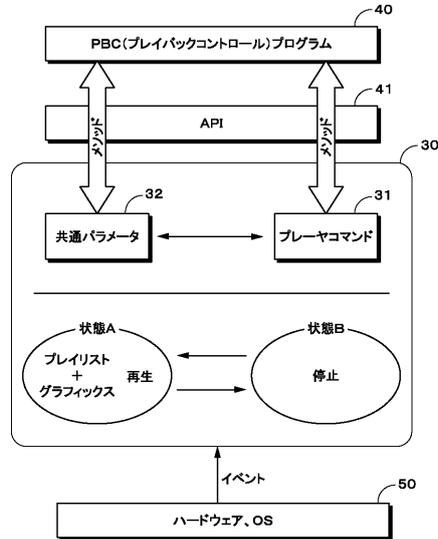
【 図 2 6 】



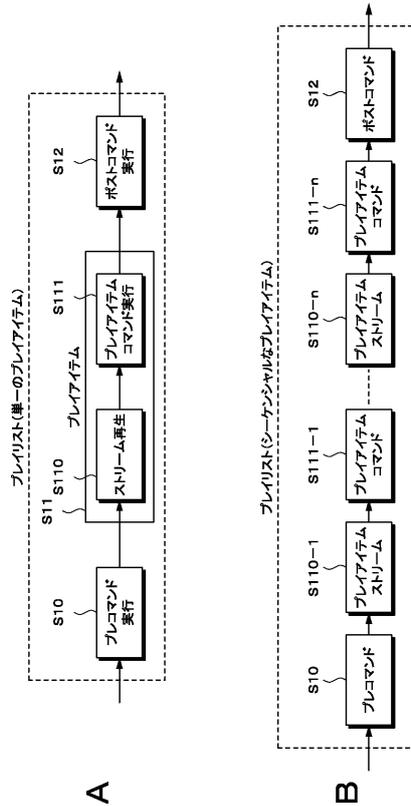
【 図 2 4 】



【 図 2 5 】



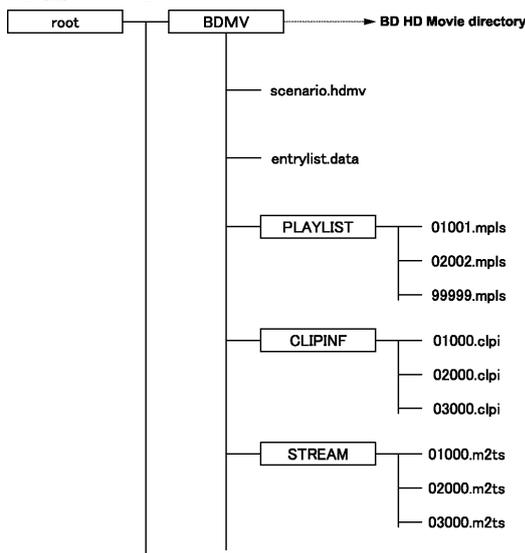
【 図 2 7 】



【 図 2 8 】

コマンド	備考
再生開始位置指定	
LinkPlaylist(playListNumber)	playListNumberで指定されたPlayListの再生を開始する。
LinkPlayItem(playListNumber, playItemNumber)	指定のPlayListの指定のPlayItemから再生を開始する。playItemNumberはPlayItem_idであり、0から始まる。PlayListの先頭から再生するのであれば、playItemNumberは0になる。
Link(position)(object) position="prev" "next" "top" "Parent" "tail" object=(PlayList PlayItem Chapter)	シナリオ内移動。現在再生中の箇所から前後のPlayList, PlayItem, Chapterに移動する。
Exit	シナリオの再生を停止する。標準レジスタの値は保持されない。
RSM	前回再生を停止した箇所から再生を再開する。保存されているresume情報を呼び出し、レジスタにセットして再生を開始する。
Player状態の取得	
getMenuDescriptionLanguage()	メニュー表示の際の言語を取得
getScenarioNumber()	再生中のシナリオ番号を得る。
getPlayListNumber()	再生中のプレイリスト番号を得る。
getChapterNumber()	再生中のチャプター番号を得る。
getPlayerSupport()	プレーヤのバージョン、有する機能を取得。
Videoストリームに関するもの	
getVideoStreamAvailability()	指定のビデオストリームが含まれているかを得る。
setVideoStreamNumber()	デコードするビデオストリームを指定する。
getVideoStreamNumber()	選択中のビデオストリームの番号を得る。
getVideoStreamAttribute()	ビデオストリームの属性を得る。(符号化方式、解像度、アスペクト比、4:3時のディスプレイモード、ClosedCaption)
setAngleNumber()	アングル番号を指定する。
getAngleNumber()	選択中のアングル番号を得る。
getMaxVideoStreams()	選択可能なビデオストリームの数を得る。
	getVideoStreamAvailability()があれば十分か？

【 図 2 9 】



【 図 3 0 】

シンタクス	データ長 (ビット)	Mnemonic
scenario.hdmv{		
type_indicator	8*4	bslbf
version_number	8*4	bslbf
scenario_start_address	32	
reserved_for_future_use	224	bslbf
Autoplay()		
for(i=0;i<N1;i++){		
padding_word	16	bslbf
}		
Scenario()		
for(i=0;i<N2;i++){		
padding_word	16	bslbf
}		
}		

【 図 3 1 】

シンタクス	データ長 (ビット)	Mnemonic
Autoplay(){		
length	32	uimsbf
reserved	16	
number_of_commands	16	
for(i=0;i<number_of_commands;i++){		
command(i)	32	
}		uimsbf
}		

【 図 3 2 】

シンタクス	データ長 (ビット)	Mnemonic
Scenario(){		
length	32	
flags	32	
number_of_PlayLists	16	
for(i=0;i<Number_of_PlayLists;i++){		
Pre_Command_start_id	32	
Post_Command_start_id	32	
number_of_Pre_Commands	32	
number_of_Post_Commands	32	
reserved	32	
number_of_PlayItems	32	
for(PlayItem_id=0;		
PlayItem_id<number_of_PlayItems;		
PlayItem_id++){		
PI_Command_start_id	32	
number_of_PI_Commands	32	
}		
reserved		
// Command table for each PlayList		
number_of_PL_Commands	16	
for(j=0;j<number_of_PL_Commands;j++){		
PL_Command(j)	32	
}		
}		
}		

【 図 3 3 】

シンタクス	データ長 (ビット)	Mnemonic
entrylist.data{		
type_indicator	8*4	bslbf
version_number	8*4	bslbf
ScenarioEntry_start_address	32	uimsbf
reserved_for_future_use	224	bslbf
AppInfo()		
for(i=0;i<N1;i++){		
padding_word	16	bslbf
}		
ScenarioEntry()		
for(i=0;i<N2;i++){		
padding_word	16	bslbf
}		
}		

【 図 3 4 】

シタクス	データ長 (ビット)	Mnemonic
AppInfo(){		
length	32	uimsbf
reserved_for_future_use	16	bslbf
HDMV_name_character_set	8	bslbf
reserved_for_word_align	7	bslbf
PIN_valid_flag	1	bslbf
PIN	8*4	bslbf
// UOP_mask_table() // For directory	64	
HDMV_name_length	8	uimsbf
HDMV_name	8*255	bslbf
}		

【 図 3 5 】

シタクス	データ長 (ビット)	Mnemonic
ScenarioEntry(){		
length	32	unimsbf
name_character_set	8	bslbf
// Entry PL for the Top Menu		
Top Menu PL() {		
flags	32	bslbf
TopMenu_ref_to_PlayList_file_name	8*10	bslbf
TopMenu_ref_to_PlayItem_id	16	unimsbf
TopMenu_name_length	8	unimsbf
TopMenu_name	8*255	bslbf
}		
// Title Entries		
number_of_Titles	16	unimsbf
for(unit title_number=0;		
title_number<Number_of_Titles;		
title_number++){		
flags	32	bslbf
Title_ref_to_PlayList_file_name	8*10	bslbf
Title_ref_to_PlayItem_id	16	unimsbf
Title_name_length	8	unimsbf
Title_name	8*255	bslbf
}		
// Stream Setup Menu for each PL		
number_of_PlayLists	16	unimsbf
for(i=0;i<Number_of_PlayLists;i++){		
SSMenu_flags	32	bslbf
SSMenu_ref_to_PlayList_file_name	8*10	bslbf
SSMenu_ref_to_PlayItem_id	16	unimsbf
}		
}		

【 図 3 8 】

PL_playback_type	Meaning
0x0	リザーブ領域
0x1	プレイアイテムをシーケンシャル再生する (通常再生)
0x2	プレイアイテムをランダム再生する
0x3	プレイアイテムをシャッフル再生する

【 図 3 9 】

PL_random_access_mode	Meaning
0x0	飛び込み再生・変速再生許可
0x1	飛び込み再生・変速再生禁止

【 図 4 0 】

シタクス	データ長 (ビット)	Mnemonic
PlayList() {		
length	32	unimsbf
number_of_PlayItems	16	unimsbf
number_of_SubPlayItems	16	unimsbf
for(PlayItem_id=0;		
PlayItem_id<number_of_PlayItems;		
PlayItem_id++){		
PlayItem()		
}		
for(SubPlayItem_id=0;		
SubPlayItem_id<number_of_SubPlayItems;		
SubPlayItem_id++){		
SubPlayItem()		
}		
}		

【 図 3 6 】

シタクス	データ長 (ビット)	Mnemonic
xxxx.mpls{		
type_indicator	8*4	bslbf
version_number	8*4	bslbf
PlayList_start_address	32	unimsbf
PlayListMark_start_address	32	unimsbf
reserved_for_future_use	192	bslbf
PLControlInfo()		
for(i=0;i<N1;i++){		
padding_word	16	bslbf
}		
PlayList()		
for(i=0;i<N2;i++){		
padding_word	16	bslbf
}		
PlayListMark()		
for(i=0;i<N3;i++){		
padding_word	16	bslbf
}		
}		

【 図 3 7 】

シタクス	データ長 (ビット)	Mnemonic
PLControlInfo(){		
length	32	unimsbf
reserved_for_future_use	8	bslbf
PlayList_character_set	8	unimsbf
reserved_for_future_use	8	
PL_playback_type	8	
if(PL_playback_type==0x2		
PL_playback_type==0x3) {		
playback_count	16	
} else {		
reserved_for_word_align	16	
}		
PL_UOP_mask_table() // For PlayList	64	
reserved_for_word_align	8	
PL_random_access_mode	8	
reserved_for_word_align	8	bslbf
PlayList_duration	4*6	bslbf
PlayList_name_length	8	unimsbf
PlayList_name	8*255	bslbf
PlayList_detail_length	16	unimsbf
PlayList_detail	8*1200	bslbf
}		

【 図 4 1 】

シタクス	データ長 (ビット)	Mnemonic
PlayItem(){		
length	16	uimsbf
reserved_for_word_align	8	bslbf
Clip_information_file_name	8*5	bslbf
Clip_codec_identifier	8*4	bslbf
reserved_for_future_use	7	bslbf
is_multi_angle	1	bslbf
reserved_for_future_use	4	bslbf
connection_condition	4	uimsbf
ref_to_STC_id	8	uimsbf
IN_time	32	uimsbf
OUT_time	32	uimsbf
PI_UOP_mask_table()	64	bslbf
PID filter()		
reserved_for_word_align	8	bslbf
PI_random_access_mode	8	uimsbf
reserved_for_word_align	8	bslbf
still_mode	8	uimsbf
if(still_mode==0x1) {		
still_time	16	uimsbf
} else {		
reserved_for_word_align	16	bslbf
}		
// Angle		
if(is_multi_angle) {		
number_of_angles	8	uimsbf
is_seamless_angle_change	8	uimsbf
for(angle_id=1;		
angle_id<number_of_angles;angle_id++){		
Clip_information_file_name	8*5	bslbf
ref_to_STC_id	8	uimsbf
IN_time	32	uimsbf
OUT_time	32	uimsbf
}		
}		
}		

【 図 4 2 】

PI_random_access_mode	Meaning
0x0	飛び込み再生・変速再生許可
0x1	飛び込み再生・変速再生禁止

【 図 4 3 】

still_mode	Meaning
0x0	Stillなし。
0x1	有限時間のStill。時間は次のstill_timeで指定。
0x2	無限時間のStill。ユーザが解除するまでStillを続ける。
0x3-0xf	reserved

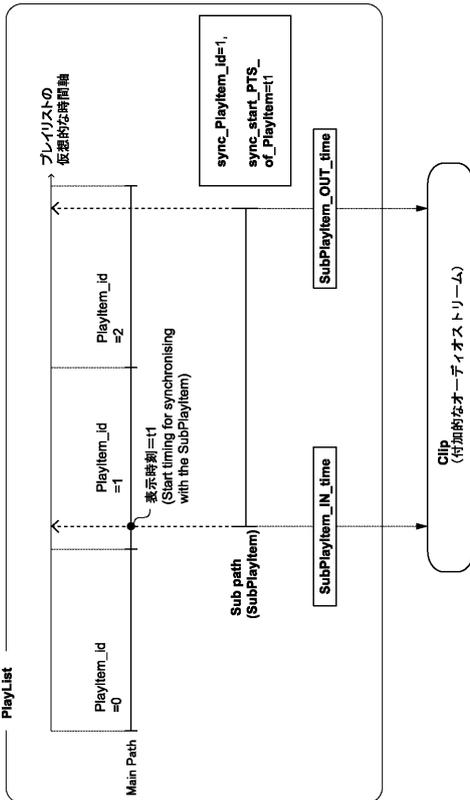
【 図 4 4 】

is_seamless_angle_change	Meaning
0x0	ノンシームレス切り替えのアングル
0x1	シームレス切り替え可能なアングル

【 図 4 5 】

シンタクス	データ長 (ビット)	Mnemonic
SubPlayItem() {		
length	16	unimbsf
Clip_information_file_name	8*5	bslbf
Clip_codec_identifier	8*4	bslbf
reserved_for_future_use	7	bslbf
is_repeat_flag	1	bslbf
SubPlayItem_type	8	bslbf
ref_to_STC_id	8	unimbsf
SubPlayItem_IN_time	32	unimbsf
SubPlayItem_OUT_time	32	unimbsf
if(is_repeat_flag=0) {		
sync_PlayItem_id	16	unimbsf
sync_start_PTS_of_PlayItem	32	unimbsf
} else {		
reserved_for_word_align	16	
reserved_for_word_align	32	
}		
}		

【 図 4 7 】



【 図 4 6 】

is_repeat_flag	Meaning
0	main pathと同期再生する。
1	main pathと同期再生しない。再生を繰り返す。

【 図 4 8 】

シンタクス	データ長 (ビット)	Mnemonic
zzzz.clip {		
type_indicator	8*4	bslbf
version_number	8*4	bslbf
SequenceInfo_start_address	32	uimbsf
ProgramInfo_start_address	32	uimbsf
CPI_start_address	32	uimbsf
ClipMark_start_address	32	uimbsf
reserved_for_future_use	128	bslbf
ClipInfo()		
for(i=0;<N1;i++) {		
padding_word	16	bslbf
}		
SequenceInfo()		
for(i=0;<N2;i++) {		
padding_word	16	bslbf
}		
ProgramInfo()		
for(i=0;<N3;i++) {		
padding_word	16	bslbf
}		
CPI()		
for(i=0;<N4;i++) {		
padding_word	16	bslbf
}		
ClipMark()		
for(i=0;<N5;i++) {		
padding_word	16	bslbf
}		
}		

【 図 4 9 】

シンタクス	データ長 (ビット)	Mnemonic
ClipInfo() {		
length	32	unimbsf
reserved	8	bslbf
application_type	8	unimbsf
Clip_stream_type	8	unimbsf
reserved	40	unimbsf
TS_recording_rate	32	unimbsf
num_of_source_packets	32	unimbsf
BD_system_use	1024	bslbf
TS_type_info_block()		
}		

【 図 5 0 】

application_type	meaning
0	対応するm2tsファイルが、HDMVトランスポートストリームのルールに従っていない。
1	対応するm2tsファイルが、HDMVトランスポートストリームのルールに従っている。(通常のHDMVストリーム)
2	対応するm2tsファイルが、オーディオ再生に同期する静止画用のHDMVトランスポートストリームのルールに従っている。(タイムベーススライドショー)
3	対応するm2tsファイルが、オーディオとは非同期に再生される静止画用のHDMVトランスポートストリームのルールに従っている。(ブラウザプルスライドショー)

【 図 5 1 】

シンタクス	データ長 (ビット)	Mnemonic
SequenceInfo() {		
length	32	unimbsf
reserved for word align	8	bslbf
num_of_ATC_sequences	8	unimbsf
for(atc_id=0;atc_id<num_of_ATC_sequences;atc_id++) {		
SPN_ATC_start [atc_id]	32	unimbsf
num_of_STC_sequences [atc_id]	8	unimbsf
offset_STC_id [atc_id]	8	unimbsf
for(stc_id=offset_STC_id[atc_id];		
stc_id<(num_of_STC_sequences[atc_id]		
+offset_STC_id[atc_id]);		
stc_id++) {		
PCR_PID [atc_id] [stc_id]	16	unimbsf
SPN_STC_start [atc_id] [stc_id]	32	unimbsf
presentation_start_time [atc_id] [stc_id]	32	unimbsf
presentation_end_time [atc_id] [stc_id]	32	unimbsf
}		
}		
}		

【 図 5 2 】

シンタクス	データ長 (ビット)	Mnemonic
ProgramInfo() {		
length	32	unimbsf
reserved for word align	8	bslbf
num_of_program_sequences	8	unimbsf
for(i=0;i<num_of_program_sequences;i++) {		
SPN_program_sequence_start [i]	32	unimbsf
program_map_PID [i]	16	bslbf
num_of_streams_in_ps [i]	8	unimbsf
num_of_groups [i]	8	unimbsf
for(stream_index=0;		
stream_index<num_of_streams_in_ps[i];		
stream_index++) {		
stream_PID [i] [stream_index]		
StreamCodingInfo(i,stream_index)	16	unimbsf
}		
}		
}		

【 図 5 3 】

シンタクス	データ長 (ビット)	Mnemonic
StreamCodingInfo(i,stream_index) {		
length	8	bslbf
stream_coding_type	8	unimbsf
if(stream_coding_type==0x02) {		
video_format	4	unimbsf
frame_rate	4	unimbsf
aspect_ratio	4	unimbsf
reserved_for_word_align	2	bslbf
cc_flag	1	unimbsf
reserved_for_word_align	1	bslbf
} else if(stream_coding_type==0x80 //		
stream_coding_type==0x81 //		
stream_coding_type==0x82) {		
audio_presentation_type	4	unimbsf
sampling_frequency	4	unimbsf
language_code	16	bslbf
reserved_for_word_align	8	bslbf
} else if(stream_coding_type==0x90) {		
language_code	16	bslbf
T.B.D		
} else if(stream_coding_type==0xA0) {}		
language_code	16	bslbf
T.B.D		
}		
}		

【 図 5 4 】

シンタクス	データ長 (ビット)	Mnemonic
CPI() {		
length	32	unimbsf
reserved_for_word_align	12	bslbf
CPI_type	4	unimbsf
EP_map_for_HDMV()		
}		

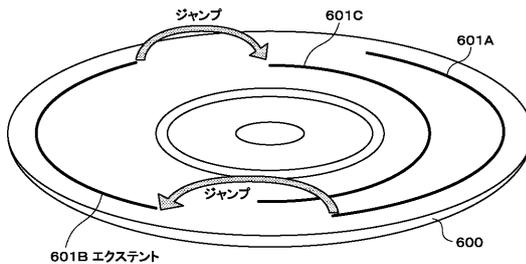
【 図 5 5 】

CPI_type	Meaning
0	reserved for future use
1	EP_map type
2	TU_map type
3-7	reserved for future use
8	EP_map type for HDMV
9-15	reserved for future use

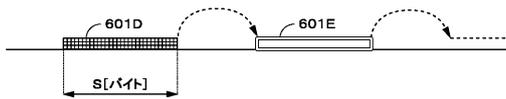
【 図 5 6 】

シンタクス	データ長 (ビット)	Mnemonic
EP_map_for_one_stream_PID(EP_stream_type,Nc,Nf) {		
EP_fine_table_start_address	32	unimbsf
for(i=0;i<Nc;i++) {		
ref_to_EP_fine_id [i]	18	unimbsf
PTS_EP_coarse [i]	14	unimbsf
SPN_EP_coarse [i]	32	unimbsf
}		
for(i=0;i<Nf;i++) {		
padding_word	16	bslbf
}		
for(EP_fine_id=0;EP_fine_id<Nf;EP_fine_id++) {		
is_angle_change_point [EP_fine_id]	1	bslbf
l_end_position_offset [EP_fine_id]	3	bslbf
PTS_EP_fine [EP_fine_id]	11	unimbsf
SPN_EP_fine [EP_fine_id]	17	unimbsf
}		
}		

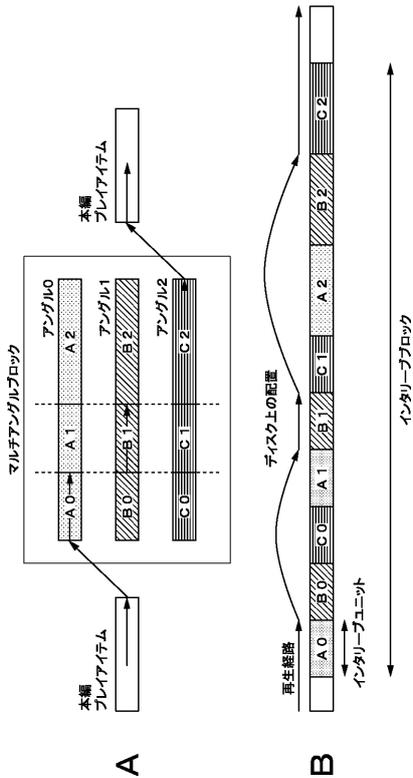
【 図 5 7 】



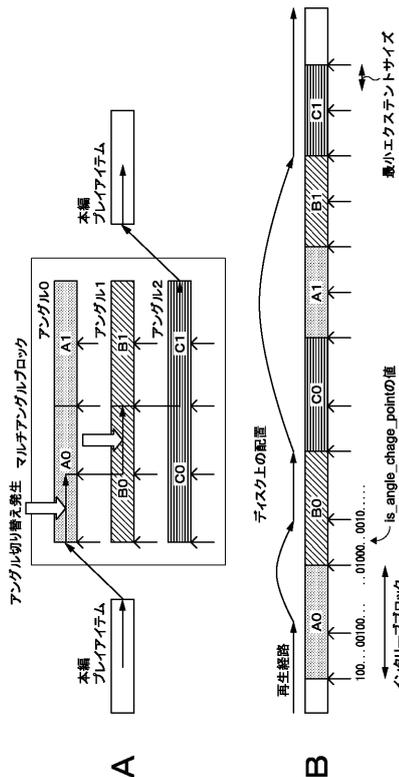
【 図 5 8 】



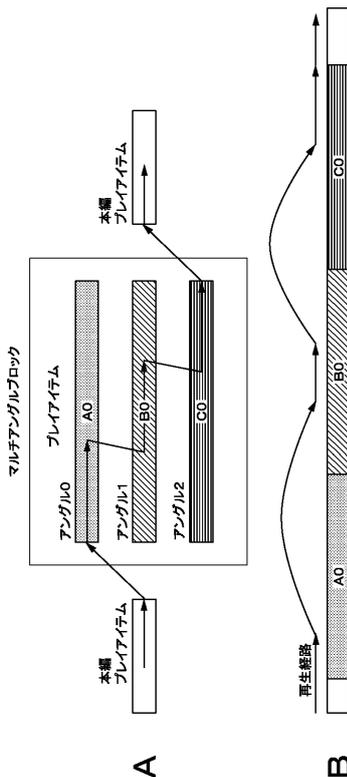
【 図 5 9 】



【 図 6 0 】



【 図 6 2 】



【 図 6 1 】

is_angle_change_point	Meaning
0	このEPエントリがアングル切り替え可能点に該当しない。
1	このEPエントリがアングル切り替え可能点に該当する。

【手続補正書】

【提出日】平成24年7月17日(2012.7.17)

【手続補正1】

【補正対象書類名】特許請求の範囲

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項1】

1または複数のフレームを単位としてエンコードされているビデオストリームと、該エンコード単位毎に設けられた、該ビデオストリームのアングルを該エンコード単位の先頭で切り替え可能か否かを示す識別情報とを記録媒体に格納し、

上記識別情報に基づく位置で、上記アングル毎にインタリーブされて記録された上記ビデオストリームの再生位置を変更して該アングルを構成するビデオストリームを読み出すことを可能にさせるデータ格納方法。

【請求項2】

請求項1に記載のデータ格納方法であって、

上記識別情報が上記アングルの終端側の所定領域に設けられている

データ格納方法。

【請求項3】

請求項2に記載のデータ格納方法であって、

上記所定領域の大きさは、第1の領域から第2の領域への最大アクセス時間と、上記ビデオストリームの読み出し速度と該ビデオストリームの再生速度との差分とに基づく

データ格納方法。

【手続補正3】

【補正対象書類名】明細書

【補正対象項目名】0001

【補正方法】変更

【補正の内容】

【0001】

この発明は、ブルーレイディスク(Blu-ray Disc)といった大容量の記録媒体に記録されたプログラムに対するユーザによるインタラクティブな操作を可能とするデータ格納方法に関する。

【手続補正4】

【補正対象書類名】明細書

【補正対象項目名】0013

【補正方法】変更

【補正の内容】

【0013】

この発明は、上述した課題を解決するために、1または複数のフレームを単位としてエンコードされているビデオストリームと、エンコード単位毎に設けられた、ビデオストリームのアングルをエンコード単位の先頭で切り替え可能か否かを示す識別情報とを記録媒体に格納し、識別情報に基づく位置で、アングル毎にインタリーブされて記録されたビデオストリームの再生位置を変更してアングルを構成するビデオストリームを読み出すことを可能にさせるデータ格納方法である。

フロントページの続き

(51)Int.Cl.

F I

テーマコード(参考)

G 1 1 B 20/12 1 0 3

Fターム(参考) 5C053 FA24 GB02 GB06 GB08 GB11 GB12 GB38 JA01 JA22 LA06
5D044 AB05 AB07 BC02 CC04 DE17 DE27 DE49 EF05 GK12