

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号
特許第4715633号
(P4715633)

(45) 発行日 平成23年7月6日 (2011.7.6)

(24) 登録日 平成23年4月8日 (2011.4.8)

(51) Int.Cl.	F I
H O 4 N 5/92 (2006.01)	H O 4 N 5/92 H
G 1 1 B 20/10 (2006.01)	G 1 1 B 20/10 3 O 1 Z
G 1 1 B 20/12 (2006.01)	G 1 1 B 20/12
G 1 1 B 27/00 (2006.01)	G 1 1 B 27/00 D
G 1 1 B 27/10 (2006.01)	G 1 1 B 27/10 A
請求項の数 14 (全 53 頁) 最終頁に続く	

(21) 出願番号	特願2006-140839 (P2006-140839)	(73) 特許権者	000002185
(22) 出願日	平成18年5月19日 (2006.5.19)		ソニー株式会社
(65) 公開番号	特開2007-312246 (P2007-312246A)		東京都港区港南1丁目7番1号
(43) 公開日	平成19年11月29日 (2007.11.29)	(74) 代理人	100082762
審査請求日	平成21年4月2日 (2009.4.2)		弁理士 杉浦 正知
		(72) 発明者	森本 直樹
			東京都品川区北品川6丁目7番35号 ソ
			ニー株式会社内
		(72) 発明者	有留 憲一郎
			東京都品川区北品川6丁目7番35号 ソ
			ニー株式会社内
		(72) 発明者	磯部 幸雄
			東京都品川区北品川6丁目7番35号 ソ
			ニー株式会社内
		最終頁に続く	

(54) 【発明の名称】 記録装置、記録方法および記録プログラム、ならびに、編集装置、編集方法および編集プログラム

(57) 【特許請求の範囲】

【請求項 1】

ビデオデータを圧縮符号化して記録媒体に記録する記録装置において、
少なくとも1枚の独立的に復号が可能なピクチャを含む複数枚のピクチャからなるグループを単位とし、予測符号化によるフレーム間圧縮を用いてビデオデータを圧縮符号化する符号化部と、

上記符号化部で圧縮符号化されたビデオデータを記録媒体に記録するのを制御する記録制御部と、

上記ビデオデータに対して上記グループ単位のアクセス位置を示すマーク情報を生成するマーク情報生成部と

を有し、

上記マーク情報生成部は、
再生順で直前のグループの情報を用いずに復号可能なグループに対して上記アクセス位置を指定するときには、上記指定されるアクセス位置から該グループの再生順の先頭に遡った位置を示す上記マーク情報を生成する
記録装置。

【請求項 2】

請求項 1 に記載の記録装置において、

上記マーク情報生成部は、

再生順で直前のグループの情報を用いて復号を行うグループに対して上記アクセス位置

を指定するときには、該グループに属する上記独立的に復号が可能なピクチャの位置を示す上記マーク情報を生成する記録装置。

【請求項 3】

請求項 2 に記載の記録装置において、

指定しようとする上記アクセス位置の含まれるグループが再生順で直前のグループの情報をを用いずに復号可能なグループであるか否かを判別する判別部をさらに有し、

上記マーク情報生成部は、上記判別部による判別結果に応じて、上記グループの再生順の先頭位置を示す上記マーク情報を生成するか、上記グループに属する上記独立的に復号が可能なピクチャの位置を示す上記マーク情報を生成するかを決定する記録装置。

10

【請求項 4】

請求項 3 に記載の記録装置において、

上記マーク情報生成部は、

記録開始時に生成される上記グループの先頭の位置を示す上記マーク情報を自動的に生成する記録装置。

【請求項 5】

請求項 4 に記載の記録装置において、

ユーザの操作を受け付け該操作に応じてアクセス位置を指定する操作部をさらに有し、上記マーク情報生成部は、

上記操作部に対する操作により指定されたアクセス位置が記録開始位置以外の位置を示していれば該アクセス位置が含まれる上記グループに属する上記独立的に復号が可能なピクチャの位置を示す上記マーク情報を生成する記録装置。

20

【請求項 6】

請求項 5 に記載の記録装置において、

上記ビデオデータの再生方法を示す再生制御情報を生成する再生制御情報生成部をさらに有し、

上記マーク情報は、上記再生制御情報に格納される記録装置。

30

【請求項 7】

ビデオデータを圧縮符号化して記録媒体に記録する記録方法において、

少なくとも 1 枚の独立的に復号が可能なピクチャを含む複数枚のピクチャからなるグループを単位とし、予測符号化によるフレーム間圧縮を用いてビデオデータを圧縮符号化する符号化のステップと、

上記符号化のステップにより圧縮符号化されたビデオデータを記録媒体に記録するのを制御する記録制御のステップと、

上記ビデオデータに対して上記グループ単位のアクセス位置を示すマーク情報を生成するマーク情報生成のステップと

40

を有し、

上記マーク情報生成のステップは、

再生順で直前のグループの情報をを用いずに復号可能なグループに対して上記アクセス位置を指定するときには、上記指定されるアクセス位置から該グループの再生順の先頭に遡った位置を示す上記マーク情報を生成する

記録方法。

【請求項 8】

ビデオデータを圧縮符号化して記録媒体に記録する記録方法をコンピュータ装置に実行させる記録プログラムにおいて、

上記記録方法は、

50

少なくとも 1 枚の独立的に復号が可能なピクチャを含む複数枚のピクチャからなるグループを単位とし、予測符号化によるフレーム間圧縮を用いてビデオデータを圧縮符号化する符号化のステップと、

上記符号化のステップにより圧縮符号化されたビデオデータを記録媒体に記録するのを制御する記録制御のステップと、

上記ビデオデータに対して上記グループ単位のアクセス位置を示すマーク情報を生成するマーク情報生成のステップと

を有し、

上記マーク情報生成のステップは、

再生順で直前のグループの情報を用いずに復号可能なグループに対して上記アクセス位置を指定するときには、上記指定されるアクセス位置から該グループの再生順の先頭に遡った位置を示す上記マーク情報を生成する
記録プログラム。

10

【請求項 9】

ビデオデータに対してアクセス位置を指定するマーク情報を生成する編集装置において、

少なくとも 1 枚の独立的に復号が可能なピクチャを含む複数枚のピクチャからなるグループを単位とし、予測符号化によるフレーム間圧縮を用いて圧縮符号化されたビデオデータを復号する復号部と、

上記ビデオデータに対して上記グループ単位のアクセス位置を示すマーク情報を生成するマーク情報生成部と、

20

を有し、

上記マーク情報生成部は、

再生順で直前のグループの情報を用いずに復号可能なグループに対して上記アクセス位置を指定するときには、上記指定されるアクセス位置から該グループの再生順の先頭に遡った位置を示す上記マーク情報を生成する
編集装置。

【請求項 10】

請求項 9 に記載の編集装置において、

上記マーク情報生成部は、

再生順で直前のグループの情報をを用いて復号を行うグループに対して上記アクセス位置を指定するときには、該グループに属する上記独立的に復号が可能なピクチャの位置を示す上記マーク情報を生成する
編集装置。

30

【請求項 11】

請求項 9 または 10 に記載の編集装置において、

上記アクセス位置の含まれるグループが再生順で直前のグループの情報をを用いずに復号可能なグループであるか否かを判別する判別部をさらに有し、

上記マーク情報生成部は、上記判別部による判別結果に応じて、上記指定されるアクセス位置から該グループの再生順の先頭に遡った位置を示す上記マーク情報を生成するか、
上記グループに属する上記独立的に復号が可能なピクチャの位置を示す上記マーク情報を生成するかを決定する

40

編集装置。

【請求項 12】

ビデオデータに対してアクセス位置を指定するマーク情報を生成する編集方法において、

少なくとも 1 枚の独立的に復号が可能なピクチャを含む複数枚のピクチャからなるグループを単位とし、予測符号化によるフレーム間圧縮を用いて圧縮符号化されたビデオデータを復号する復号のステップと、

上記ビデオデータに対して上記グループ単位のアクセス位置を示すマーク情報を生成す

50

るマーク情報生成のステップと、
を有し、

上記マーク情報生成のステップは、
再生順で直前のグループの情報を用いずに復号可能なグループに対して上記アクセス位置を指定するときには、上記指定されるアクセス位置から該グループの再生順の先頭に遡った位置を示す上記マーク情報を生成する
編集方法。

【請求項 13】

ビデオデータに対してアクセス位置を指定するマーク情報を生成する編集方法をコンピュータ装置に実行させる編集プログラムにおいて、

上記編集方法は、

少なくとも1枚の独立的に復号が可能なピクチャを含む複数枚のピクチャからなるグループを単位とし、予測符号化によるフレーム間圧縮を用いて圧縮符号化されたビデオデータを復号する復号のステップと、

上記ビデオデータに対して上記グループ単位のアクセス位置を示すマーク情報を生成するマーク情報生成のステップと、
を有し、

上記マーク情報生成のステップは、

再生順で直前のグループの情報を用いずに復号可能なグループに対して上記アクセス位置を指定するときには、上記指定されるアクセス位置から該グループの再生順の先頭に遡った位置を示す上記マーク情報を生成する
編集プログラム。

【請求項 14】

ビデオデータを圧縮符号化して記録媒体に記録する記録装置において、

少なくとも1枚の独立的に復号が可能なピクチャを含む複数枚のピクチャからなるグループを単位とし、予測符号化によるフレーム間圧縮を用いてビデオデータを圧縮符号化する符号化部と、

上記符号化部で圧縮符号化されたビデオデータを記録媒体に記録するのを制御する記録制御部と、

上記ビデオデータに対して上記グループ単位のアクセス位置を示すマーク情報を生成するマーク情報生成部と
を有し、

上記マーク情報生成部は、

再生順で直前のグループの情報を用いて復号を行うグループに対して上記アクセス位置を指定するときには、該グループに属する上記独立的に復号が可能なピクチャの位置を示す上記マーク情報を生成する
記録装置。

【発明の詳細な説明】

【技術分野】

【0001】

この発明は、A V (Audio/Video) ストリームデータからなるクリップに対する編集処理を行うようにした記録装置、記録方法および記録プログラム、ならびに、編集装置、編集方法および編集プログラムに関する。

【背景技術】

【0002】

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

10

20

30

40

50

【特許文献1】特開2004-350251

【0003】

このような記録可能な記録媒体においては、記録媒体に記録されたAVデータは、編集可能であることが望ましい。AVデータに対する代表的な編集方法の例としては、所定単位のAV(Audio/Video)データを分割編集することが挙げられる。ビデオデータおよびオーディオデータは、一般的には多重化された所定のデータ単位として記録される。

【0004】

また、ビデオデータは、データ容量が膨大となるため、所定の方式で圧縮符号化されて記録媒体に記録されるのが一般的である。ビデオデータの圧縮符号化方式としては、MPEG2(Moving Picture Experts Group 2)方式が標準的な方式として知られている。また、近年では、MPEG2の圧縮符号化方式をさらに推し進めて効率的な符号化を行えるようにした、ITU-T(International Telecommunication Union-Telecommunication Standardization Sector)勧告H.264あるいはISO(International Organization for Standardization)/IEC(International Electrotechnical Commission)国際標準14496-10(MPEG-4パート10)Advanced Video Coding(以下、H.264|AVCと略称する)も、普及してきている。

【0005】

これらMPEG2およびH.264|AVCにおいては、直交変換などを用いたフレーム内符号化を行うと共に、動き補償を用いた予測符号化によるフレーム間符号化をさらにを行い、圧縮率を高めている。以下、MPEG2を例にとって、予測符号化によるフレーム間圧縮について説明する。

【0006】

先ず、MPEG2のデータストリーム構造について、概略的に説明する。MPEG2は、動き補償予測符号化と、DCTによる圧縮符号化とを組み合わせたものである。MPEG2のデータ構造は、階層構造をなしており、下位から、ブロック層、マクロブロック層、スライス層、ピクチャ層、GOP層およびシーケンス層となっている。ブロック層は、DCTを行う単位であるDCTブロックからなる。マクロブロック層は、複数のDCTブロックで構成される。スライス層は、ヘッダ部と、1以上のマクロブロックより構成される。ピクチャ層は、ヘッダ部と、1以上のスライスとから構成される。ピクチャは、1画面に対応する。各層の境界は、それぞれ所定の識別符号で識別可能なようになっている。

【0007】

GOP層は、ヘッダ部と、フレーム内符号化に基づくピクチャであるI(Intra-coded)ピクチャと、予測符号化に基づくピクチャであるP(Predictive-coded)ピクチャB(Bi-directionally predictive coded)ピクチャとから構成される。Iピクチャは、それ自身の情報のみでデコードが可能であり、PおよびBピクチャは、基準画像として前あるいは前後の画像が必要とされ、単独ではデコードされない。例えばPピクチャは、自身より時間的に前のIピクチャまたはPピクチャを基準画像として用いてデコードされる。また、Bピクチャは、自身の前後のIピクチャまたはPピクチャの2枚のピクチャを基準画像として用いてデコードされる。最低1枚のIピクチャを含むそれ自身で完結したグループをGOP(Group Of Picture)と呼び、MPEGのストリームにおいて独立してアクセス可能な最小の単位とされる。

【0008】

GOPは、1または複数のピクチャから構成される。以下では、GOPは、複数のピクチャから構成されるものとする。GOPにおいて、GOP内で完全にデコードが可能で、GOPで閉じた構造を持つクローズドGOPと、デコードの際に符号化順で1つ前のGOPの情報を参照することができるオープンGOPとの2種類がある。オープンGOPは、クローズドGOPと比較して、より多くの情報を参照してデコードできるため高画質を得られ、一般的に用いられている。

【0009】

図37を用いて、フレーム間圧縮を行ったデータのデコード処理について説明する。こ

10

20

30

40

50

ここでは、1 GOP が 1 枚の I ピクチャ、4 枚の P ピクチャおよび 10 枚の B ピクチャの、計 15 枚のピクチャから構成されるものとする。GOP 内の I、P および B ピクチャの表示順は、図 37A に一例が示されるように、「 $B_0 B_1 I_2 B_3 B_4 P_5 B_6 B_7 P_8 B_9 B_{10} P_{11} B_{12} B_{13} P_{14}$ 」のようになる。なお、添え字は表示順を示す。

【0010】

この例では、最初の 2 枚の B_0 ピクチャおよび B_1 ピクチャは、1 つ前の GOP における最後尾の P_{14} ピクチャと、この GOP 内の I_2 ピクチャを用いて予測されデコードされたピクチャである。GOP 内の最初の P_5 ピクチャは、 I_2 ピクチャから予測されデコードされたピクチャである。他の P_8 ピクチャ、 P_{11} ピクチャおよび P_{14} は、それぞれ 1 つ前の P ピクチャを用いて予測されデコードされたピクチャである。また、I ピクチャ以降の各 B ピクチャは、それぞれ前後の I および / または P ピクチャから予測されデコードされたピクチャである。

10

【0011】

一方、B ピクチャは、時間的に前後の I または P ピクチャを用いて予測されデコードされるため、ストリームや記録媒体上における I、P および B ピクチャの並び順は、デコーダにおけるデコードの順序を考慮して決める必要がある。すなわち、B ピクチャをデコードするための I および / または P ピクチャは、当該 B ピクチャよりも常に先にデコードされていなければならない。

【0012】

上述の例では、ストリームや記録媒体上の各ピクチャの配列は、図 37B に例示されるように、「 $I_2 B_0 B_1 P_5 B_3 B_4 P_8 B_6 B_7 P_{11} B_9 B_{10} P_{14} B_{12} B_{13}$ 」のようになり、この順でデコーダに入力される。なお、添え字は、図 37A に対応し、表示順を示す。

20

【0013】

デコーダにおけるデコード処理は、図 37C に示されるように、まず I_2 ピクチャをデコードし、デコードされたこの I_2 ピクチャと 1 つ前の GOP における最後尾（表示順）の P_{14} ピクチャとにより B_0 ピクチャおよび B_1 ピクチャを予測しデコードする。そして、 B_0 ピクチャおよび B_1 ピクチャをデコードされた順にデコーダから出力し、次に I_2 ピクチャを出力する。 B_1 ピクチャが出力されると、次に P_5 ピクチャが I_2 ピクチャを用いて予測されデコードされる。そして、 I_2 ピクチャおよび P_5 ピクチャを用いて B_3 ピクチャおよび B_4 ピクチャが予測されデコードされる。そして、デコードされた B_3 ピクチャおよび B_4 ピクチャをデコードされた順にデコーダから出力し、次に P_8 ピクチャを出力する。

30

【0014】

以下、同様にして、B ピクチャの予測に用いる P または I ピクチャを B ピクチャより先にデコードし、このデコードされた P または I ピクチャを用いて B ピクチャを予測してデコードし、デコードされた B ピクチャを出力してから、当該 B ピクチャをデコードするために用いた P または I ピクチャを出力する処理が繰り返される。記録媒体上やストリームにおける図 37B のようなピクチャ配列は、一般的に用いられるものである。

【発明の開示】

【発明が解決しようとする課題】

【0015】

40

このように、GOP 構造で以てフレーム間圧縮を行ったビデオデータに対してアクセス位置を指定することにより、ビデオデータを分割編集しようとする場合、GOP の境界を指定して分割する第 1 の方法と、GOP 内の I ピクチャを指定して分割する第 2 の方法とが考えられる。第 1 の方法によれば、I ピクチャ以降のピクチャは、図 37 を用いて説明したように、この I ピクチャを参照して順次、デコードできるので、再生時に、編集結果に基づき分割されたビデオデータの先頭へアクセスした際の、ジャンプ先のフレームの表示に遅延が少ない利点がある。

【0016】

一方、この第 1 の方法では、分割指定された境界位置と GOP の境界とが一致しないことになる。そのため、分割されたビデオデータの先頭へアクセスした場合、GOP の先頭

50

のピクチャから再生することができないという問題点があった。

【0017】

また、第1の方法によれば、分割されたビデオデータの後端部においても、後端とGOP境界とが一致しない。そのため、例えば分割されたビデオデータの後端のピクチャが含まれるGOPが先行するGOPとは異なるシーンとして生成されたGOPである場合、再生時に、当該分割されたビデオデータの後端部において、シーンの繋がりの無い映像が数フレーム分、再生されてしまうという問題点があった。

【0018】

第2の方法では、GOPの先頭と編集により分割指定された境界とが一致するので、上述の第1の方法の場合のように、GOPの先頭側において再生されないピクチャが発生したり、分割されビデオデータの後端部においてシーンの繋がりの無い映像が再生されてしまう問題は発生しない。

【0019】

しかしながら、この第2の方法では、分割されたビデオデータの先頭のフレームが含まれるGOPがオープンGOPである場合、当該GOPの表示順がIピクチャより前のBピクチャは、図37を用いて説明したように、当該GOPのIピクチャと、直前のGOPにおける最後のPピクチャとを用いてデコードされる。そのため、当該分割されたビデオデータにジャンプされた際には、当該GOPの直前のGOPの最後のPピクチャをもデコードする必要があり、ジャンプ操作に応じた映像出力の遅延が大きくなってしまうという問題点があった。

【0020】

このように、従来では、GOP構造で以てフレーム間圧縮を行ったビデオデータに対して分割する編集を行う場合に、再生映像に問題が発生せず、ユーザの利便性を損なわないような方法が存在しなかったという問題点があった。

【0021】

したがって、この発明の目的は、GOP構造を有したビデオデータに対して分割編集を行う際のユーザの利便性を向上させることができるような記録装置、記録方法および記録プログラム、ならびに、編集装置、編集方法および編集プログラムを提供することにある。

【課題を解決するための手段】

【0022】

上述した課題を解決するために、第1の発明は、ビデオデータを圧縮符号化して記録媒体に記録する記録装置において、少なくとも1枚の独立的に復号が可能なピクチャを含む複数枚のピクチャからなるグループを単位とし、予測符号化によるフレーム間圧縮を用いてビデオデータを圧縮符号化する符号化部と、符号化部で圧縮符号化されたビデオデータを記録媒体に記録するのを制御する記録制御部と、ビデオデータに対してグループ単位のアクセス位置を示すマーク情報を生成するマーク情報生成部とを有し、マーク情報生成部は、再生順で直前のグループの情報をを用いずに復号可能なグループに対してアクセス位置を指定するときには、指定されるアクセス位置からグループの再生順の先頭に遡った位置を示すマーク情報を生成する記録装置である。

【0023】

また、第2の発明は、ビデオデータを圧縮符号化して記録媒体に記録する記録方法において、少なくとも1枚の独立的に復号が可能なピクチャを含む複数枚のピクチャからなるグループを単位とし、予測符号化によるフレーム間圧縮を用いてビデオデータを圧縮符号化する符号化のステップと、符号化のステップにより圧縮符号化されたビデオデータを記録媒体に記録するのを制御する記録制御のステップと、ビデオデータに対してグループ単位のアクセス位置を示すマーク情報を生成するマーク情報生成のステップとを有し、マーク情報生成のステップは、再生順で直前のグループの情報をを用いずに復号可能なグループに対してアクセス位置を指定するときには、指定されるアクセス位置から該グループの再生順の先頭に遡った位置を示すマーク情報を生成する記録方法である。

【 0 0 2 4 】

また、第3の発明は、ビデオデータを圧縮符号化して記録媒体に記録する記録方法をコンピュータ装置に実行させる記録プログラムにおいて、記録方法は、少なくとも1枚の独立的に復号が可能なピクチャを含む複数枚のピクチャからなるグループを単位とし、予測符号化によるフレーム間圧縮を用いてビデオデータを圧縮符号化する符号化のステップと、符号化のステップにより圧縮符号化されたビデオデータを記録媒体に記録するのを制御する記録制御のステップと、ビデオデータに対してグループ単位のアクセス位置を示すマーク情報を生成するマーク情報生成のステップとを有し、マーク情報生成のステップは、再生順で直前のグループの情報をを用いずに復号可能なグループに対してアクセス位置を指定するときには、指定されるアクセス位置からグループの再生順の先頭に遡った位置を示すマーク情報を生成する記録プログラムである。

10

【 0 0 2 5 】

また、第4の発明は、ビデオデータに対してアクセス位置を指定するマーク情報を生成する編集装置において、少なくとも1枚の独立的に復号が可能なピクチャを含む複数枚のピクチャからなるグループを単位とし、予測符号化によるフレーム間圧縮を用いて圧縮符号化されたビデオデータを復号する復号部と、ビデオデータに対してグループ単位のアクセス位置を示すマーク情報を生成するマーク情報生成部とを有し、マーク情報生成部は、再生順で直前のグループの情報をを用いずに復号可能なグループに対してアクセス位置を指定するときには、指定されるアクセス位置からグループの再生順の先頭に遡った位置を示すマーク情報を生成する編集装置である。

20

【 0 0 2 6 】

また、第5の発明は、ビデオデータに対してアクセス位置を指定するマーク情報を生成する編集方法において、少なくとも1枚の独立的に復号が可能なピクチャを含む複数枚のピクチャからなるグループを単位とし、予測符号化によるフレーム間圧縮を用いて圧縮符号化されたビデオデータを復号する復号のステップと、ビデオデータに対してグループ単位のアクセス位置を示すマーク情報を生成するマーク情報生成のステップとを有し、マーク情報生成のステップは、再生順で直前のグループの情報をを用いずに復号可能なグループに対してアクセス位置を指定するときには、指定されるアクセス位置からグループの再生順の先頭に遡った位置を示すマーク情報を生成する編集方法である。

30

【 0 0 2 7 】

また、第6の発明は、ビデオデータに対してアクセス位置を指定するマーク情報を生成する編集方法をコンピュータ装置に実行させる編集プログラムにおいて、編集方法は、少なくとも1枚の独立的に復号が可能なピクチャを含む複数枚のピクチャからなるグループを単位とし、予測符号化によるフレーム間圧縮を用いて圧縮符号化されたビデオデータを復号する復号のステップと、ビデオデータに対してグループ単位のアクセス位置を示すマーク情報を生成するマーク情報生成のステップとを有し、マーク情報生成のステップは、再生順で直前のグループの情報をを用いずに復号可能なグループに対してアクセス位置を指定するときには、指定されるアクセス位置から該グループの再生順の先頭に遡った位置を示すマーク情報を生成する編集プログラムである。

40

【 0 0 2 8 】

また、第7の発明は、ビデオデータを圧縮符号化して記録媒体に記録する記録装置において、少なくとも1枚の独立的に復号が可能なピクチャを含む複数枚のピクチャからなるグループを単位とし、予測符号化によるフレーム間圧縮を用いてビデオデータを圧縮符号化する符号化部と、符号化部で圧縮符号化されたビデオデータを記録媒体に記録するのを制御する記録制御部と、ビデオデータに対してグループ単位のアクセス位置を示すマーク情報を生成するマーク情報生成部とを有し、マーク情報生成部は、再生順で直前のグループの情報をを用いて復号を行うグループに対してアクセス位置を指定するときには、グループに属する独立的に復号が可能なピクチャの位置を示すマーク情報を生成する記録装置である。

【 0 0 2 9 】

50

上述したように、第1、第2および第3の発明は、少なくとも1枚の独立的に復号が可能なピクチャを含む複数枚のピクチャからなるグループを単位とし、予測符号化によるフレーム間圧縮を用いて圧縮符号化したビデオデータに対して、再生順で直前のグループの情報を用いずに復号可能なグループに対してアクセス位置を指定するときには、指定されるアクセス位置からグループの再生順の先頭に遡った位置を示すマーク情報を生成し、圧縮符号化されたビデオデータを記録媒体に記録するようにしているため、記録媒体からビデオデータを再生したときに、再生順で直前のグループの情報を用いずに復号可能なグループへのジャンプを指定した場合、ジャンプ先における表示の遅延を少なくできる。

【0030】

また、第4、第5および第6の発明は、少なくとも1枚の独立的に復号が可能なピクチャを含む複数枚のピクチャからなるグループを単位とし、予測符号化によるフレーム間圧縮を用いて圧縮符号化したビデオデータに対して、再生順で直前のグループの情報を用いずに復号可能なグループに対してアクセス位置を指定するときには、指定されるアクセス位置からグループの再生順の先頭に遡った位置を示すマーク情報を生成し、圧縮符号化されたビデオデータを記録媒体に記録するようにしているため、記録媒体からビデオデータを再生したときに、再生順で直前のグループの情報を用いずに復号可能なグループへのジャンプを指定した場合、ジャンプ先における表示の遅延を少なくできる。

【0031】

また、第7の発明は、少なくとも1枚の独立的に復号が可能なピクチャを含む複数枚のピクチャからなるグループを単位とし、予測符号化によるフレーム間圧縮を用いてビデオデータを圧縮符号化したビデオデータに対して、再生順で直前のグループの情報を用いて復号を行うグループに対してアクセス位置を指定するときには、グループに属する独立的に復号が可能なピクチャの位置を示すマーク情報を生成し、圧縮符号化されたビデオデータを記録媒体に記録するようにしているため、記録が停止された後に再び記録が開始された場合、記録の停止部分において表示すべき区間の終端とデコード区間の終端とが一致し、このプレイリストマークに基づくチャプタの終端で内容的に関連性のないフレームが表示されることがない。

【発明の効果】

【0032】

第1、第2および第3の発明は、上述したように、少なくとも1枚の独立的に復号が可能なピクチャを含む複数枚のピクチャからなるグループを単位とし、予測符号化によるフレーム間圧縮を用いて圧縮符号化したビデオデータに対して、再生順で直前のグループの情報を用いずに復号可能なグループに対してアクセス位置を指定するときには、指定されるアクセス位置からグループの再生順の先頭に遡った位置を示すマーク情報を生成し、圧縮符号化されたビデオデータを記録媒体に記録するようにしているため、記録媒体からビデオデータを再生したときに、再生順で直前のグループの情報を用いずに復号可能なグループへのジャンプを指定した場合、ジャンプ先における表示の遅延を少なくできる効果がある。

【0033】

また、第4、第5および第6の発明は、少なくとも1枚の独立的に復号が可能なピクチャを含む複数枚のピクチャからなるグループを単位とし、予測符号化によるフレーム間圧縮を用いて圧縮符号化したビデオデータに対して、再生順で直前のグループの情報を用いずに復号可能なグループに対してアクセス位置を指定するときには、指定されるアクセス位置からグループの再生順の先頭に遡った位置を示すマーク情報を生成し、圧縮符号化されたビデオデータを記録媒体に記録するようにしているため、記録媒体からビデオデータを再生したときに、再生順で直前のグループの情報を用いずに復号可能なグループへのジャンプを指定した場合、ジャンプ先における表示の遅延を少なくできる効果がある。

【0034】

また、第7の発明は、少なくとも1枚の独立的に復号が可能なピクチャを含む複数枚のピクチャからなるグループを単位とし、予測符号化によるフレーム間圧縮を用いてビデオ

10

20

30

40

50

データを圧縮符号化したビデオデータに対して、再生順で直前のグループの情報を用いて復号を行うグループに対してアクセス位置を指定するときには、グループに属する独立に復号が可能なピクチャの位置を示すマーク情報を生成し、圧縮符号化されたビデオデータを記録媒体に記録するようにしているため、記録が停止された後に再び記録が開始された場合、記録の停止部分において表示すべき区間の終端とデコード区間の終端とが一致し、このプレイリストマークに基づくチャプタの終端で内容的に関連性のないフレームが表示されることがないという効果がある。

【発明を実施するための最良の形態】

【0035】

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

【0036】

例えばI T U - T (International Telecommunication Union-Telecommunication Standardization Sector)勧告H . 2 6 4あるいはI S O (International Organization for Standardization) / I E C (International Electrotechnical Commission)国際標準1 4 4 9 6 - 1 0 (M P E G - 4 パート1 0) Advanced Video Coding (以下、H . 2 6 4 | A V Cと略称する)に規定される符号化方式で符号化され、M P E G 2 システムズに従い多重化されたビットストリームは、クリップA Vストリーム（またはA Vストリーム）と称される。クリップA Vストリームは、所定のファイルシステムによりファイルとしてディスクに記録される。このファイルを、クリップA Vストリームファイル（またはA Vストリームファイル）と称する。

【0037】

クリップA V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A V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A Vストリームファイルの一部だけを再生する仕組み、さらには、特殊再生や頭出し再生を滑らかに行うための情報などをデータベースとしてディスクに記録しておく必要がある。

【0038】

図1は、この発明に適用可能なA V C H Dフォーマットに規定されるデータモデルを概略的に示す。このA V C H Dフォーマットによれば、データ構造は、図1に示されるように4層のレイヤよりなる。最も最下層のレイヤは、クリップA Vストリームが配置されるレイヤである（便宜上、クリップレイヤと呼ぶ）。その上のレイヤは、クリップA Vストリームに対する再生箇所を指定するための、プレイリスト(PlayList)と、プレイアイテム(PlayItem)とが配置されるレイヤである（便宜上、プレイリストレイヤと呼ぶ）。さらにその上のレイヤは、プレイリストに対して再生順などを指定するコマンドからなるムービーオブジェクト(Movie Object)などが配置されるレイヤである（便宜上、オブジェクトレイヤと呼ぶ）。最上層のレイヤは、記録媒体に格納されるタイトルなどを管理するインデックステーブルが配置される（便宜上、インデックスレイヤと呼ぶ）。

【0039】

クリップレイヤについて説明する。クリップA Vストリームは、ビデオデータやオーディオデータがM P E G 2 T S (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A V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。

【0040】

また、クリップA Vストリームには、字幕を表示するグラフィックスストリームであるO

10

20

30

40

50

B ストリーム (Overlay Bitmap stream) や、メニュー表示などに用いられるデータ (ボタン画像データなど) をストリームにした M B ストリーム (Menu Bitmap stream) ストリームを多重化することができる。

【 0 0 4 1 】

クリップ A V 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 A V 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。

【 0 0 4 2 】

ファイルは、一般的に、バイト列として扱われる。クリップ A V ストリームファイルのコンテンツは、時間軸上に展開され、クリップ中のエン트리ポイントは、主に時間ベースで指定される。所定のクリップへのアクセスポイントのタイムスタンプが与えられた場合、クリップ A V 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。

10

【 0 0 4 3 】

プレイリストレイヤについて説明する。プレイリストは、再生する A V ストリームファイルの指定と、指定された A V ストリームファイルの再生箇所を指定する再生開始点 (I N 点) と再生終了点 (O U T 点) の集まりとから構成される。この再生開始点と再生終了点の情報を一組としたものは、プレイアイテム (PlayItem) と称される。プレイリストは、プレイアイテムの集合で構成される。プレイアイテムを再生するということは、そのプレイアイテムに参照される A V ストリームファイルの一部分を再生するということになる。すなわち、プレイアイテム中の I N 点および O U T 点情報に基づき、クリップ中の対応する区間が再生される。

20

【 0 0 4 4 】

オブジェクトレイヤについて説明する。ムービーオブジェクトは、ナビゲーションコマンドプログラムと、ムービーオブジェクトとを連携するターミナルインフォメーションを含む。ナビゲーションプログラムは、プレイリストの再生を制御するためのコマンド (ナビゲーションコマンド : navigation command) である。

【 0 0 4 5 】

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

30

【 0 0 4 6 】

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

40

【 0 0 4 7 】

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

【 0 0 4 8 】

図 3 は、上述のようなクリップ A V ストリーム、クリップ情報 (Stream Attributes)、

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。

【0049】

10

また、図4のように、複数のプレイリストから同一のクリップを参照することもできる。また、1のプレイリストから複数のクリップを指定することもできる。クリップは、プレイリスト中のプレイアイテムに示されるIN点およびOUT点により、参照される。図4の例では、クリップ300は、プレイリスト310のプレイアイテム320から参照されると共に、プレイリスト311を構成するプレイアイテム321および322のうちプレイアイテム321から、IN点およびOUT点で示される区間が参照される。また、クリップ301は、プレイリスト311のプレイアイテム322からIN点およびOUT点で示される区間が参照されると共に、プレイリスト312のプレイアイテム323および324のうち、プレイアイテム323のIN点およびOUT点で示される区間が参照される。図4の例では、クリップ301は、さらに別のプレイリストからも参照されている。

20

【0050】

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

【0051】

ルートディレクトリの下に、ディレクトリ"BDMV"がおかれる。さらに、必要に応じてディレクトリ"AVCHDTN"が置かれる。ディレクトリ"AVCHDTN"には、例えばクリップの代表画像を所定サイズに縮小したサムネイルファイルが置かれる。ディレクトリ"BDMV"に、図1を用いて説明したデータ構造が格納される。

30

【0052】

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

【0053】

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

40

【0054】

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

50

【 0 0 5 5 】

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

【 0 0 5 6 】

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

【 0 0 5 7 】

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

【 0 0 5 8 】

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

【 0 0 5 9 】

図6において、フィールドTypeIndicatorは、32ビットのデータ長を有し、このファイルがインデックステーブルであることを示す。フィールドTypeIndicator2は、32ビットのデータ長を有し、このファイル"index.bdmv"のバージョンを示す。フィールドIndexesStartAddressは、32ビットのデータ長を有し、このシンタクス内にあるブロックblkIndexes()の開始アドレスを示す。

【 0 0 6 0 】

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

【 0 0 6 1 】

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

【 0 0 6 2 】

ブロックblkIndexes()は、このファイル"index.bdmv"の実質的な内容であって、このフ

10

20

30

40

50

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

【 0 0 6 3 】

図 7 は、ブロックblkIndexes()の一例の構造を表すシンタクスを示す。フィールドLengthは、32ビットのデータ長を有し、このフィールドLength直後からこのブロックblkIndexes()の終わりまでのデータ長を示す。続けて、ブロックFirstPlaybackTitle()およびブロックMenuTitle()が配される。

【 0 0 6 4 】

10

ブロックFirstPlaybackTitle()は、ファーストプレイバックで用いられるオブジェクトに関する情報が記述される。ブロックFirstPlaybackTitle()は、1ビットのデータ長を有する領域reservedに続けて固定値"1"が記述される。さらに31ビットのデータ長を有する領域reservedを介して固定値"1"が記述される。そして、14ビットのデータ長を有する領域reservedを介して、16ビットのデータ長を有するフィールドFirstPlaybackTitleMobjIDRefが配される。このフィールドFirstPlaybackTitleMobjIDRefにより、ファーストプレイバックタイトルで用いられるムービーオブジェクトのIDを示す。

【 0 0 6 5 】

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

20

【 0 0 6 6 】

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

【 0 0 6 7 】

ブロックMenuTitle()は、トップメニューで用いられるオブジェクトに関する情報が記述される。ブロックMenuTitle()は、1ビットのデータ長を有する領域reservedに続けて固定値"1"が記述される。さらに31ビットのデータ長を有する領域reservedを介して固定値"1"が記述される。そして、14ビットのデータ長を有する領域reservedを介して、16ビットのデータ長を有するフィールドMenuTitleMobjIDRefが配される。フィールドMenuTitleMobjIDRefは、メニュータイトルで用いられるムービーオブジェクトのIDを示す。

30

【 0 0 6 8 】

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

40

【 0 0 6 9 】

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

【 0 0 7 0 】

図 8 は、ディレクトリ"BDMV"の直下に置かれるファイル"MovieObject.bdmv"の一例の構

50

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

【0071】

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

【0072】

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

【0073】

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

【0074】

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

【0075】

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

【0076】

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

【0077】

次の、値command_idをループ変数とするforループ文により、フィールドNumberOfNavigationCommands[mobj_id]に示される数だけ、ナビゲーションコマンドが記述される。すなわち、このforループ文中に配されるフィールドNavigationCommand[mobj_id][command_id]は、値mobj_idによって指し示されるブロックMovieObject[mobj_id]()に含まれる、値co

10

20

30

40

50

mmmand_idで示される順番のナビゲーションコマンドNavigationCommandを格納する。値command_idは、0 から始まる値で、ナビゲーションコマンドNavigationCommandは、このforループ文中に記述される順序で定義される。

【 0 0 7 8 】

図 1 0 は、プレイリストファイル"xxxxx.mpls"の一例の構造を表すシンタクスを示す。フィールドTypeIndicatorは、3 2 ビット (4 バイト) のデータ長を有し、このファイルがプレイリストファイルであることを示す。フィールドTypeIndicator2は、3 2 ビット (4 バイト) のデータ長を有し、このプレイリストファイルのバージョンを示す。フィールドPlayListStartAddressは、3 2 ビットのデータ長を有し、このシンタクス中のブロックblkPlayList()の開始アドレスを示す。

10

【 0 0 7 9 】

フィールドPlayListMarkStartAddressは、3 2 ビットのデータ長を有し、このシンタクス中のブロックblkPlayListMark()の開始アドレスを示す。フィールドExtensionDataStartAddressは、3 2 ビットのデータ長を有し、このシンタクス中のブロックblkExtensionData()の開始アドレスを示す。フィールドExtensionDataStartAddressは、ブロックblkExtensionData()の開始アドレスを、ファイル"xxxxx.mpls"の最初のバイトからの相対バイト数を表した値である。相対バイト数は、" 0 "から開始される。若し、このフィールドExtensionDataStartAddressの値が" 0 "であれば、このファイル"xxxxx.mpls"内に、ブロックblkExtensionData()が存在しないことを示す。

【 0 0 8 0 】

20

1 6 0 ビットのデータ長を有する領域reservedを介してブロックblkAppInfoPlayList()が配される。ブロックblkAppInfoPlayList()は、次のブロックblkPlayList()に記述されるプレイリストのタイプ、再生制限などの情報が記述される。ブロックblkPlayList()は、プレイリストが記述される。ブロックblkPlayListMark()は、チャプタジャンプなどでジャンプされるポイントが記述される。ブロックblkExtensionData()は、所定の拡張データを格納可能とするためのブロックである。

【 0 0 8 1 】

図 1 1 は、ブロックblkPlayList()の一例の構造を表すシンタクスを示す。フィールドLengthは、3 2 ビットのデータ長を有し、このフィールドLengthの直後からブロックblkPlayList()の最後までまでのデータ長を示す。フィールドLengthに続けて1 6 ビットのデータ長を有する領域reservedが配され、次にフィールドNumberOfPlayItemsが配される。フィールドNumberOfPlayItemsは、1 6 ビットのデータ長を有し、このブロックblkPlayList()に含まれるプレイアイテムの数を示す。フィールドNumberOfSubPathは、このブロックblkPlayList()に含まれるサブパスの数を示す。

30

【 0 0 8 2 】

次のforループ文に従い、フィールドNumberOfPlayItemsで示される数だけ、プレイアイテムが記述されるブロックblkPlayItem()が記述される。forループ文に基づくカウント数がブロックblkPlayItem()の識別子PlayItem_idとなる。さらに次のforループ文に従い、フィールドNumberOfSubPathで示される数だけ、ブロックblkSubPath()が記述される。forループ文に基づくカウント数がブロックblkSubPath()の識別子SubPath_idとなる。

40

【 0 0 8 3 】

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

【 0 0 8 4 】

図 1 2 は、ブロックblkPlayItem()の一例の構造を表すシンタクスを示す。フィールドLengthは、1 6 ビットのデータ長を有し、このフィールドLengthの直後からブロックblkPlayItem()の最後までまでのデータ長を示す。

【 0 0 8 5 】

50

フィールドClipInformationFileNameは、40ビット(5バイト)のデータ長を有し、このブロックblkPlayItem()が参照するクリップインフォメーションファイルのファイル名が示される。このプレイアイテムにおいて、フィールドClipInformationFileNameで示されるファイル名のクリップインフォメーションファイルが読み出される。フィールドClipCodecIdentifierは、32ビット(4バイト)のデータ長を有し、このブロックblkPlayItem()によるプレイアイテムにおいて用いられるクリップAVストリームのコーデック方式を示す。

【0086】

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

【0087】

フィールドConnectionConditionの値が"5"で、当該プレイアイテムが参照するクリップにおいて、オーディオデータの記録長がビデオデータの記録長に対して長くされる(図13A参照)。これにより、クリップとクリップとを接続する際に、オーディオデータのフェードアウト処理が可能とされる。例えば、ユーザによる記録停止操作によりクリップがクローズされる場合に、フィールドConnectionConditionの値が"5"とされる。以下、このフィールドConnectionConditionの値が"5"で示されるクリップの接続方法を、第1のシームレス接続と呼ぶ。

【0088】

フィールドConnectionConditionの値が"6"で、当該プレイアイテムが参照するクリップにおいて、オーディオデータの記録長がビデオデータの記録長に対して同じにされる(図13B参照)。これにより、クリップとクリップとの接続をシームレスに行うことが可能とされる。例えば、ユーザ操作に応じた記録停止以外の理由、例えばシステム要因に基づきクリップがクローズされる場合に、フィールドConnectionConditionの値が"6"とされる。以下、このフィールドConnectionConditionの値が"6"で示されるクリップの接続方法を、第2のシームレス接続と呼ぶ。

【0089】

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

【0090】

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

10

20

30

40

50

る。

【 0 0 9 1 】

なお、数値の記述において"0x"は、その数値が16進表記されていることを示す。これは、以下の同様な表記について共通である。

【 0 0 9 2 】

ブロックblkSTNTable()は、このブロックblkPlayItem()によるプレイアイテムが管理しているクリップAVストリームの属性、PID番号、記録媒体上での記録位置などが管理される。

【 0 0 9 3 】

図14は、ブロックblkPlayListMark()の一例の構造を表すシンタクスを示す。フィールドLengthは、32ビットのデータ長を有し、このフィールドLengthの直後からブロックblkPlayListMark()の最後までデータの長を示す。

10

【 0 0 9 4 】

フィールドNumberOfPlayListMarksは、16ビットのデータ長を有し、このブロックblkPlayListMark()に含まれるプレイリストマークの数を示す。次のforループ文に従い、フィールドNumberOfPlayListMarksで示される数だけプレイリストマークの情報が記述される。

【 0 0 9 5 】

forループ文内において、8ビットのデータ長を有する領域reserveに続けてフィールドMarkTypeが配される。フィールドMarkTypeは、8ビットのデータ長を有し、マークのタイプを示す。プレイリストマークには、エントリマーク(Entry Mark)およびリンクポイント(Link Point)の2タイプが定義されており、このフィールドMarkTypeにより、何れのタイプであるかが示される。チャプタを定義するためには、エントリマークを用いる。リンクポイントは、この発明と関連性が薄いので、説明を省略する。上述したフィールドNumberOfPlayListMarksは、エントリマークおよびリンクポイントを合計した値を示す。

20

【 0 0 9 6 】

フィールドRefToPlayItemIDは、16ビットのデータ長を有し、マークが打たれるプレイアイテムを参照する識別情報PlayItem_idが記述される。フィールドMarkTimeStampは、32ビットのデータ長を有し、マークが打たれるポイントを示すタイムスタンプが記述される。フィールドEntryESPIDは、16ビットのデータ長を有し、マークによって指し示されるエレメンタリストリームを含んでいるTSパケットのPIDの値を示す。フィールドDurationは、45kHzのクロックを単位とした計測による、32ビットのデータ長を有する符号無し整数である。このフィールドDurationに格納される値が"0"であれば、このフィールドDurationは、意味を成さない。

30

【 0 0 9 7 】

図15は、クリップインフォメーションファイルの一例の構造を表すシンタクスを示す。フィールドTypeIndicatorは、32ビット(4バイト)のデータ長を有し、このファイルがクリップインフォメーションファイルであることを示す。フィールドTypeIndicator2は、32ビット(4バイト)のデータ長を有し、このクリップインフォメーションファイルのバージョンを示す。

40

【 0 0 9 8 】

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

【 0 0 9 9 】

フィールドExtensionDataStartAddressは、このクリップインフォメーションファイルの最初のバイトからの相対バイト数で、ブロックblkExtensionData()の開始アドレスを示

50

す。相対バイト数は、" 0 "から開始される。若し、このフィールドExtensionDataStartAddressの値が" 0 "であれば、このファイル"index.bdmv"内に、ブロックblkExtensionData()が存在しないことを示す。

【 0 1 0 0 】

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

【 0 1 0 1 】

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

【 0 1 0 2 】

図16は、クリップインフォメーションファイルにおけるブロックblkCPI()の一例の構造を表すシンタクスを示す。MPEGストリームのような、フレーム間圧縮を行っている符号化ストリームにおいては、デコード開始可能な箇所は、GOP(Group Of Picture)の先頭など一部の箇所に限定されていることが多い。C P I(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、C P I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。

【 0 1 0 3 】

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C P I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 1 0 4 】

なお、このC P I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、E P(Entry Point)エントリと称する。

【 0 1 0 5 】

図16において、フィールドLengthは、32ビットのデータ長を有し、このフィールドLengthの直後からブロックblkCPI()の最後までデータ長を示す。次のif文に従い、フィールドLengthの値が0でなければ、12ビットのデータ長を有する領域reservedを介してフィールドCPITypeが配される。フィールドCPITypeは、4ビットのデータ長を有し、C P Iの種類を示す。次のブロックblkEPMMap()は、対応するクリップAVストリームファイルにおけるP T S値とバイトアドレスとの関連付けを行うテーブルが格納される。

【 0 1 0 6 】

図17は、ブロックblkEPMMap()の一例の構造を表すシンタクスを示す。8ビットのデータ長を有する領域reservedを介してフィールドNumberOfStreamPIDEntriesが配される。フィールドNumberOfStreamPIDEntriesは、8ビットのデータ長を有し、ブロックblkEPMMap()におけるブロックblkEPMMapForOneStreamPIDのエントリ数を示す。forループ文に従い、値[k]をループ変数として、フィールドNumberOfStreamPIDEntriesに示される数だけ、エントリポイントに関する情報が記述される。

【 0 1 0 7 】

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

【0108】

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

【0109】

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

【0110】

図18は、ブロックblkEMapForOneStreamPID(EP_stream_type, Nc, Nf)の一例の構造を表すシンタクスを示す。ブロックblkEMapForOneStreamPID(EP_stream_type, Nc, Nf)のセマンティクスを説明するために、先ず、ブロックblkEMapForOneStreamPID(EP_stream_type, Nc, Nf)に格納されるデータの元となるエントリである、エントリPTSEPStartおよびエントリSPNEPStartの意味について説明する。

【0111】

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

【0112】

図19は、エントリPTSEPCoarseおよびエントリPTSEPFineの一例のフォーマットについて示す。PTSすなわちエントリPTSEPStartは、データ長が33ビットの値である。MSBのビットを第32ビット、LSBのビットを第0ビットとすると、この図19の例では、大まかな単位で検索を行う際に用いられるエントリPTSEPCoarseは、エントリPTSEPStartの第32ビットから第19ビットまでの14ビットが用いられる。エントリPTSEPCoarseにより、解像度が5.8秒で、26.5時間までの範囲で検索が可能である。また、より精密な検索を行うためのエントリPTSEPFineは、エントリPTSEPStartの第19ビットから第9ビットまでの11ビットが用いられる。エントリPTSEPFineにより、解像度が5.7ミリ秒で、11.5秒までの範囲で検索が可能である。なお、第19ビットは、エントリPTSEPCoarseとエントリPTSEPFineとで共通して用いられる。また、LSB側の第0ビッ

10

20

30

40

50

トから第 8 ビットまでの 9 ビットは、用いられない。

【 0 1 1 3 】

図 2 0 は、エントリ SPNEPCoarse および エントリ SPNEPFine の一例のフォーマットについて示す。ソースパケット番号すなわちエントリ SPNEPStart は、データ長が 3 2 ビットの値である。M S B のビットを第 3 1 ビット、L S B のビットを第 0 ビットとすると、この図 2 0 の例では、大まかな単位で検索を行う際に用いられるエントリ SPNEPCoarse は、エントリ SPNEPStart の第 3 1 ビットから第 0 ビットまでの全てのビットが用いられる。また、より精密な検索を行うためのエントリ SPNEPFine は、エントリ SPNEPStart の第 1 6 ビットから第 0 ビットまでの 1 7 ビットが用いられる。エントリ SPNEPFine により、例えば略 2 5 M B (Mega Byte) の A V ストリームファイルまでの範囲で、検索が可能である。

10

【 0 1 1 4 】

なお、ソースパケット番号の場合でも、エントリ SPNEPCoarse として M S B 側の所定ビット数の値だけ用いるようにしてもよい。例えば、エントリ SPNEPCoarse として、エントリ SPNEPStart の第 3 1 ビットから第 1 6 ビットまでの 1 7 ビットを用い、エントリ SPNEPFine は、エントリ SPNEPStart の第 1 6 ビットから第 0 ビットまでの 1 7 ビットを用いる。

【 0 1 1 5 】

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

【 0 1 1 6 】

エントリ PTSEPStart は、図 1 9 で示したように、データ長が 3 3 ビットの符号無し整数であり、A V ストリーム中で、ランダムアクセスが可能なピクチャ (例えば I D R (Instantaneous Decoding Refresh) ピクチャや I (Intra) ピクチャ) から開始するビデオアクセスユニットの 3 3 ビット長の P T S を示す。

20

【 0 1 1 7 】

エントリ SPNEPStart は、図 2 0 で示したように、3 2 ビットの符号無し整数であり、エントリ PTSEPStart に関連付けられたビデオアクセスユニットの第 1 バイト目を含むソースパケットの、A V ストリームの中でのアドレスを示す。エントリ SPNEPStart は、ソースパケット番号の単位で表され、A V ストリームファイル中の最初のソースパケットから、値 " 0 " を初期値として、ソースパケット毎に 1 ずつ増加する値としてカウントされる。

【 0 1 1 8 】

図 1 8 を参照し、ブロック blkEMapForOneStreamPID (EP_stream_type, Nc, Nf) は、第 1 の for ループ文により大まかな単位での検索を行うためのサブテーブル (EP coarse table) が記述され、第 2 の for ループ文によりサブテーブル (EP coarse table) の検索結果に基づきより詳細な検索を行うためのサブテーブル (EP fine table) が記述される。

30

【 0 1 1 9 】

第 1 の for ループ文の直前に、フィールド EPFineTableStartAddress が配される。フィールド EPFineTableStartAddress は、3 2 ビットのデータ長を有し、最初の第 2 の for ループにおけるフィールド ReservedEPFine [EP_fine_id] の第 1 バイト目の開始アドレスを、ブロック blkEMapForOneStreamPID (EP_stream_type, Nc, Nf) の第 1 バイト目からの相対バイト数で示す。相対バイト数は、値 " 0 " から開始する。

40

【 0 1 2 0 】

第 1 の for ループ文は、ループ変数 [i] で以て、サブテーブル (EP coarse table) のエントリ数 Nc まで繰り返され、エントリ数 Nc の組数だけフィールド RefToEPFineID [i]、エントリ PTSEPCoarse [i] および エントリ SPNEPCoarse [i] が格納される。第 1 の for ループ文において、フィールド RefToEPFineID [i] は、1 8 ビットのデータ長を有し、フィールド RefToEPFineID [i] に続くフィールド PTSEPCoarse [i] が示すエントリ PTSEPCoarse に関連付けられるエントリ PTSEPFine を持つ、サブテーブル (EP fine table) 内のエントリ番号を示す。エントリ PTSEPFine と、このエントリ PTSEPFine に関連付けられるエントリ PTSEPCoarse とは、同一のエントリ PTSEPStart から導かれる。フィールド RefToEPFineID [i] は、第 2 の for ループ文中で記述される順番で定義されるループ変数 [EP_fine_id] の値により与えられる

50

。

【 0 1 2 1 】

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

10

【 0 1 2 2 】

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

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

【 0 1 2 3 】

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

20

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

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

【 0 1 2 4 】

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

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

【 0 1 2 5 】

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

30

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

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

【 0 1 2 6 】

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

【 0 1 2 7 】

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、プレイリストに対してプレイリストマークを設けてクリップの分割編集を行う際に、プレイリストマークを設ける位置に対応するGOPがクローズドGOPであれば、アドレス情報すなわちGOP境界にプレイリストマークを設け、当該GOPがオープンGOPであれば、再生時刻情報すなわち上述したエントリPTSEPCoarse(図20参照)で示される位置にプレイリストマークを設ける。

40

【 0 1 2 8 】

例えば、記録開始位置は、それ以前のGOPの情報をを用いることができないので、通常はクローズドGOP構造となる。そのため、記録開始位置では、プレイリストマークをGOP境界に設ける。記録開始位置以外では、プレイリストマークをエントリPTSEPCoarseで示される位置に設ける。また、記録開始位置以外でも、プレイリストマークを設けようとする位置に対応するGOPがクローズドGOPおよびオープンGOPのうち何れであるかが判別できれば、判別結果に応じて、GOP境界またはエントリPTSEPCoarseのうち適合する方にプレイリストマークを設ける。

50

【0129】

このように制御することで、チャプタジャンプ時に、ジャンプ先でのデコード範囲を最小限とすることが可能となると共に、チャプタ境界と本来の記録開始位置とが一致するようにできる。

【0130】

なお、この発明の実施の一形態では、チャプタ分割位置を指定するためのプレイリストマークは、全てタイプがエントリマークであるものとする。

【0131】

以下、この発明の実施の一形態によるチャプタ分割について、より詳細に説明する。なお、以下では、プレイリストマークの種類がエントリマークに固定的であるものとして説明する。

10

【0132】

まず、チャプタは、ユーザから見える最小の再生区間単位であって、隣接するプレイリストマークの間の区間を指す。ここで、プレイリストマークは、プレイリストで示される最後尾の位置には設けなくてもよいとされる。また、プレイリストで示される先頭の位置は、必ずプレイリストマークが設けられる。

【0133】

さらに、この実施の一形態では、分割編集の時間的な粒度は、GOP単位とする。また、記録開始位置が含まれるGOP以外のGOPは、オープンGOPであるものとする。これは、より高画質を実現するためである。

20

【0134】

まず、プレイリストに対してプレイリストマークを挿入する処理について説明する。図21は、上述したこの発明に適用可能なフォーマットにおける、プレイリスト以下の一例のファイル構造を示す。この図21の例では、プレイリスト#10は、それぞれ値PlayItem_id(図11参照)で参照される3つのプレイアイテム#0、#1および#2を含み、プレイアイテム#0および#1がクリップ#20の区間をそれぞれ参照し、プレイアイテム#2がクリップ#31の区間を参照している。また、プレイリスト#11は、2つのプレイアイテム#0および#1を含み、これらプレイアイテム#0および#1がクリップ#31の区間をそれぞれ参照している。

【0135】

30

上述したように、プレイアイテムは、時刻情報を用いて、クリップを構成するクリップインフォメーションに含まれるブロックblkEPMa()に記述されるエントリポイントを参照し、対応するクリップAVストリームファイル内のアドレスを指定する。これにより、プレイリストからクリップAVストリームファイルの部分を指定して所定の順序で再生させることができる。

【0136】

さらに、図21の例では、プレイリスト#10において、4個のプレイリストマークPLM#0、PLM#1、PLM#2およびPLM#3が設けられている。プレイリストにおいて、クリップ内のジャンプ可能なアクセス位置を示すマークを設定することができる。このマークを、プレイリストマーク(PlaylistMark: PLMと適宜略称する)と呼ぶ。これらプレイリストマークは、プレイリストファイル中のブロックblkPlaylistMark()において値PL_mark_idにより参照され、フィールドMarkTimeStampにより時刻情報で示される(図14参照)。

40

【0137】

次に、図22を用いて、プレイリストを用いたクリップの分割編集について、概略的に説明する。図22Aは、プレイリストによってクリップが参照される様子を示す。記録媒体上に記録されたクリップ#1およびクリップ#2は、それぞれ、プレイリスト#1内のプレイアイテム#1およびプレイアイテム#2から再生開始点(IN点)および再生終了点(OUT点)を指定されて参照される。この図22の例では、プレイアイテム#1は、クリップ#1の先頭(位置a)をIN点とし、後端(位置b)をOUT点としてクリップ

50

1を参照している。プレイアイテム # 2についても同様に、クリップ # 2の先頭（位置 c）および後端（位置 d）をそれぞれ I N点および O U T点として参照している。プレイリスト # 1は、プレイアイテム # 1およびプレイアイテム # 2の再生順を指定する。

【 0 1 3 8 】

したがって、この図 2 2 Aの例では、プレイリスト # 1により、クリップ # 1が位置 aから位置 bまで再生され、クリップ # 1の位置 bに続けてクリップ # 2が位置 cから再生され、位置 dで再生が終了する。すなわち、この図 2 2 Aの例では、プレイリスト # 1の指示により、クリップ # 1の先頭からクリップ # 2の最後までが連続的に再生される。

【 0 1 3 9 】

また、図 2 2 Aの例では、クリップ # 1の先頭の位置 aとクリップ # 2の先頭の位置 cとに、プレイリストマーク P L M # 1および P L M # 2がそれぞれ設定されている。この場合、例えばプレイリスト # 1に従いクリップ # 1の位置 aから位置 bの間の任意の位置で再生しているときにジャンプが指定されると、現在の再生位置に対して再生方向に向けて直近に設定されたプレイリストマークの位置（位置 aから位置 bに向けて再生している場合、プレイリストマーク P L M # 2の位置）に、再生位置がジャンプされ、プレイリストマーク P L M # 2の位置から再生が継続されることになる。

【 0 1 4 0 】

クリップの途中でプレイリストマークを設定することで、当該クリップを仮想的に分割することができる。例えば、図 2 2 Bに一例が示されるように、プレイリスト # 1に対し、クリップ # 2の所望の位置にプレイリストマーク P L M # 3を設ける。これにより、図 2 2 Cに一例が示されるように、クリップ # 2のプレイリストマーク P L M # 3に対応する位置 eをジャンプ位置として指定することが可能となり、クリップ # 2が位置 eで仮想的に分割されたものと見なすことができるようになる。

【 0 1 4 1 】

なお、ユーザ側から見た場合、プレイリストで示される区間は、チャプタを単位として再生される。チャプタは、プレイリストに対してプレイリストマークを設けることで設定することができる。例えば、プレイリストで示される区間の先頭のみプレイリストマークが設けられているか、プレイリストにプレイリストマークが全く設けられていなければ、当該プレイリスト全体が 1のチャプタを構成する。

【 0 1 4 2 】

また、上述した図 2 2 Aの例のように、クリップ # 1の先頭の位置 aに対応する位置にプレイリストマーク P L M # 1が設けられ、クリップ # 2の先頭の位置 cに対応する位置にプレイリストマーク P L M # 2が設けられていれば、プレイリストマーク P L M # 1から P L M # 2までの区間がチャプタ # 1とされ、プレイリストマーク P L M # 2からプレイリスト # 1の最後の位置までの区間がチャプタ # 2とされる。この場合は、プレイリスト # 1に示される全長がプレイリストマーク # 2により 2のチャプタに分割されたことになる。

【 0 1 4 3 】

さらに、上述した図 2 2 Cの例のように、クリップ # 2の途中の所望の位置 eに対応するプレイリストマーク P L M # 3が設けられた場合、図 2 2 Aのチャプタ # 2がプレイリストマーク P L M # 3の位置でさらに分割され、プレイリストマーク P L M # 2からプレイリストマーク P L M # 3までの区間がチャプタ # 2とされ、プレイリストマーク P L M # 3からプレイリスト # 1の最後の位置までの区間がチャプタ # 3とされる。

【 0 1 4 4 】

図 2 3は、プレイリストマークを追加挿入する一例の処理を示すフローチャートである。この処理では、プレイリスト内でプレイリストマークを格納するブロック blkPlayListMark()（図 1 4 参照）を検索して、現在再生中の位置と既に設定されているプレイリストマークとの位置関係を求め、その結果に基づきプレイリストマークを追加挿入すると共に、ブロック blkPlayListMark()の情報を書き換える。

【 0 1 4 5 】

10

20

30

40

50

ステップS 5 0 でカウント値 i を値 0 として初期化し、次のステップS 5 1 で現在再生中のプレイアイテム（図 2 3 では P I と表記）の番号が i 番目のプレイリストマーク（図 2 3 では P L M と表記）に参照されるプレイアイテムの番号より大きいか否かが判断される。

【 0 1 4 6 】

例えば、図 1 1 のブロックblkPlayList()において、ループ変数PlayItem_idによるforループ文で記述されるブロックblkPlayItem()のうち、現在再生中のプレイアイテムに対応する値PlayItem_idを求め、求められた値PlayItem_idと、ブロックblkPlayListMark()における i 番目の値RefToPlayItemIDとが比較される。若し、現在再生中のプレイアイテムの番号の方が大きいと判断されれば、カウント値 i が 1 だけインクリメントされて処理がステップS 5 1 に戻される。

10

【 0 1 4 7 】

一方、ステップS 5 1 において、現在再生中のプレイアイテムの番号が i 番目のプレイリストマークに参照されるプレイアイテムの番号より大きくない、すなわち、現在再生中のプレイアイテムと i 番目のプレイリストマークに参照されるプレイアイテムの番号とが等しいか、若しくは、現在再生中のプレイアイテムの番号よりも i 番目のプレイリストマークに参照されるプレイアイテムの番号の方が大きいと判断されると、処理はステップS 5 2 に移行される。

【 0 1 4 8 】

ステップS 5 2 では、現在再生中のプレイアイテムの番号と、 i 番目のプレイリストマークに参照されるプレイアイテムの番号とが等しいか否かが判断される。若し、等しいと判断されれば、処理は次のステップS 5 3 に移行される。一方、等しくないと判断されれば、処理は後述するステップS 5 4 に移行される。

20

【 0 1 4 9 】

ステップS 5 3 では、現在の再生時刻が i 番目のプレイリストマークの指し示す再生時刻よりも後であるか否かが判断される。現在の再生時刻は、例えばビデオストリームをデコードするデコーダから取得することができる。また、 i 番目のプレイリストマークの指し示す再生時刻は、図 1 4 を参照し、ブロックblkPlayListMark()における i 番目の値MarkTimeStampに基づく。

【 0 1 5 0 】

若し、現在の再生時刻が i 番目のプレイリストマークの指し示す時刻よりも後であると判断されたら、カウント値 i が 1 だけインクリメントされて、処理がステップS 5 3 に戻される。一方、現在の再生時刻が i 番目のプレイリストマークの指し示す再生時刻より後ではない、すなわち、現在の再生時刻が i 番目のプレイリストマークの指し示す再生時刻と等しいか、若しくは、現在の再生時刻が i 番目のプレイリストマークが指し示す再生時刻より前であれば、処理はステップS 5 4 に移行される。

30

【 0 1 5 1 】

ステップS 5 4 では、 i 番目の位置にプレイリストマーク（この場合はエントリマーク）を追加挿入する処理が行われる。ここでは、ブロックblkPlayListMark()において、既存の i 番目のプレイリストマークと、 $(i - 1)$ 番目のプレイリストマークとの間に、プレイリストマークが追加挿入されることになる。既存の i 番目のプレイリストマークは、プレイリストマークの追加挿入処理後には、 $(i + 1)$ 番目のプレイリストマークとされる。

40

【 0 1 5 2 】

より詳細には、ブロックblkPlayListMark()において、現在再生中のプレイアイテムに対応する値PlayItem_idを、フィールドRefToPlayItemIDの値に設定する。また、フィールドMarkTimeStampの値は、現在の再生時刻を示す値とされる。さらに、フィールドNumberOfPlayListMarksの値が 1 だけインクリメントされ、フィールドLengthの値が再計算される。

【 0 1 5 3 】

50

なお、実際には、ブロックblkPlayListMark()はプレイリストファイルの一部なので、図10に一例が示されるプレイリストファイルのシンタックスにおいて、フィールドExtensionDataStartAddressの値が、ブロックblkPlayListMark()におけるフィールドLengthの値に応じて書き換えられることになる。

【0154】

図24を用いてより具体的に説明する。なお、図24Aおよび図24Bは、上述した図21と同様の構造であるので、図そのものの詳細な説明は省略する。ここでは、図24AにおいてプレイリストマークPLM#2およびPLM#3の間を再生中に、位置aに対してプレイリストマークを追加挿入する場合について説明する。

【0155】

カウント値iが初期化されて値が"0"とされ、位置aに対応するプレイアイテムは、プレイアイテム#2であるので、現在再生中のプレイアイテムの番号が2とされる。一方、i番目すなわち0番目のプレイリストマーク#0は、プレイアイテム#0を参照しており、i番目のプレイリストマークが参照するプレイアイテム番号が0とされる。ステップS51の判断に基づき、現在再生中のプレイアイテムの番号がi番目のプレイリストマークが参照するプレイアイテム番号よりも大きいので、カウント値iが1だけインクリメントされて値が"1"とされ、再びステップS51の判断がなされる。

【0156】

2度目のステップS51の判断では、i番目すなわち1番目のプレイリストマークPLM#1が参照するプレイアイテム番号と、現在再生中のプレイアイテム番号とが比較される。1番目のプレイリストマークPLM#1は、プレイアイテム#1を参照しており、参照するプレイアイテム番号が1とされる。ステップS51の判断に基づき、現在再生中のプレイアイテムの番号がi番目のプレイリストマークが参照するプレイアイテム番号よりも大きいので、カウント値iが1だけインクリメントされて値が"2"とされ、再びステップS51の判断がなされる。

【0157】

3度目のステップS51の判断において、i番目すなわち2番目のプレイリストマークPLM#2は、プレイアイテム#1に指定される区間内にあり、2番目のプレイリストマークPLM#2が参照するプレイアイテム番号が1とされる。したがって、現在再生中のプレイアイテム番号が、i番目のプレイリストマークが参照するプレイアイテム番号よりも大きいとされ、カウント値iが1だけインクリメントされて値が"3"とされ、再びステップS51の判断がなされる。

【0158】

4度目のステップS51の判断において、i番目すなわち3番目のプレイリストマークPLM#3は、プレイアイテム#2に指定される区間内にあり、3番目のプレイリストマークが参照するプレイアイテム番号が"2"とされる。したがって、現在再生中のプレイアイテム番号がi番目のプレイリストマークが参照するプレイアイテム番号と等しいとされ、処理はステップS52に移行される。そして、ステップS52では、3番目のプレイリストマークPLM#3が参照するプレイアイテム番号と、現在再生中のプレイリストマークの番号とが等しいので、処理はステップS53に移行される。

【0159】

カウント値iは"3"なので、i番目のプレイリストマークであるプレイリストマークPLM#3の設定された時刻が現在再生中の位置aの時刻よりも後となる。したがって、ステップS53では、現在の再生時刻を示す値がi番目のプレイリストマークの設定された時刻を示す値よりも小さいとされ、処理はステップS54に移行されて位置aに対応するプレイリストマークが設定される。

【0160】

すなわち、参照するプレイアイテム番号が"2"で、位置aの再生時刻を示すプレイリストマークが3番目のプレイリストマークPLM#3として追加挿入される。また、プレイリストマーク数が1だけインクリメントされ、このプレイリストマークPLM#3が追加

10

20

30

40

50

挿入される直前の、当該プレイリストマーク P L M # 3 が示す再生時刻以降に設けられたプレイリストマークの番号がそれぞれ 1 だけインクリメントされる。すなわち、図 2 4 の例では、図 2 4 A に示されるプレイリストマーク P L M # 3 は、プレイリストマークの追加挿入処理により図 2 4 B に例示されるようにプレイリストマーク P L M # 4 とされ、追加挿入されたプレイリストマークが、プレイリストマーク P L M # 3 とされる。

【 0 1 6 1 】

なお、ステップ S 5 2 において、現在再生中のプレイアイテム番号と i 番目のプレイリストマークの参照するプレイアイテム番号とが等しくない場合、換言すれば、現在再生中のプレイアイテム番号が i 番目のプレイリストマークの参照するプレイアイテム番号よりも小さい場合とは、例えば、それぞれ対応する区間にプレイリストマークが設けられている 2 つのプレイアイテムの間にある、プレイリストマークが設けられていないプレイアイテムに、新たにプレイリストマークを追加挿入する場合が考えられる。

10

【 0 1 6 2 】

一例として、図 2 5 に例示されるように、プレイアイテム # 0、プレイアイテム # 1 およびプレイアイテム # 2 の 3 つのプレイアイテムを含むプレイリスト # 1 2 を考える。ここで、プレイアイテム # 0 の先頭に対応する位置と、プレイアイテム # 2 で示される区間内に、プレイリストマーク P L M # 0 およびプレイリストマーク P L M # 1 がそれぞれ設定されているものとする。

【 0 1 6 3 】

このような状態において、プレイアイテム # 1 で示される区間内の位置 b にプレイリストマークを追加挿入する場合、ステップ S 5 1 の判断では、カウント値 i が " 1 " のときに、現在再生中のプレイアイテム番号が " 1 "、i 番目のプレイリストマークが参照しているプレイアイテム番号が " 2 " とされ、現在再生中のプレイアイテム番号よりも i 番目のプレイリストマークが参照しているプレイアイテム番号の方が大きくなり、処理がステップ S 5 2 に移行される。そして、ステップ S 5 2 では、現在再生中のプレイアイテム番号が " 1 "、i 番目のプレイリストマークが参照しているプレイアイテム番号が " 2 " なので、両者が等しくないとされる。またこの場合、位置 b に対応するプレイアイテム # 1 には、他のプレイリストマークが存在しないので、ステップ S 5 3 による、プレイアイテム内での前後を判断する必要がない。したがって、処理はステップ S 5 2 からステップ S 5 4 に移行され、位置 b に対応するプレイリストマークの追加挿入処理がなされる。

20

30

【 0 1 6 4 】

次に、この発明の主旨に関わる、プレイリストマークを設ける位置の決定方法について説明する。既に説明したように、エントリ PTSEPStart で示される位置にプレイリストマークを設ける第 1 の方法と、G O P 境界の位置にプレイリストマークを設ける第 2 の方法との 2 通りの方法が考えられる。以下に、それぞれの場合における利点および欠点、ならびに、改善方法について説明する。

【 0 1 6 5 】

なお、エントリ PTSEPStart で示される位置は、例えば M P E G 2 方式のように、I ピクチャ、P ピクチャおよび B ピクチャにより G O P が構成される場合は、I ピクチャの位置に対応させることができる。また、例えば M P E G 4 や H . 2 6 4 | A V C 方式のように、I D R ピクチャをさらに用いる場合には、I D R ピクチャの位置に対応させることができる。

40

【 0 1 6 6 】

図 2 6 は、第 1 および第 2 の方法におけるそれぞれのチャプタ境界の例を示す。プレイリストマーク P L M # 1 およびプレイリストマーク P L M # 2 によりチャプタの境界が示され、プレイリストマーク P L M # 1 および P L M # 2 の間がチャプタ # 2 とされる。図 2 6 A は、第 1 の方法における一例のチャプタ境界を示す。記録が開始される位置は、G O P の先頭になるので、エントリ PTSEPStart で示される位置にプレイリストマークを設けると、G O P の先頭とプレイリストマークに基づくチャプタの先頭との間に、図 2 6 A に例示されるように、不一致区間が発生する。

50

【 0 1 6 7 】

したがって、チャプタ先頭部の不一致区間の1乃至複数フレームが再生されないことになる。例えば、記録開始位置にプレイリストマークを設けようとした場合、記録開始位置から開始されるGOPの、最初のIピクチャ(MPEG2方式の場合)より表示順で前の、例えば1乃至複数のBピクチャが再生されない不具合が生じる。

【 0 1 6 8 】

また、チャプタの後端部においては、不一致区間の1乃至複数フレームは、不一致区間の直前のGOPとは異なるGOPに属する。そのため、当該不一致区間の1乃至複数フレームは、当該不一致区間の直前のフレームと内容的に関連性が無いことがあり得る。例えば、不一致区間の直前のGOPは、記録停止に応じて記録されたGOPであって、不一致区間を含むGOPは、次の若しくは別の記録開始で記録されたGOPである場合などが考えられる。このような場合、チャプタの後端部において、シーンの的に繋がりの無いフレームが数フレーム、再生されてしまう不具合が生じる。

10

【 0 1 6 9 】

図26Bは、第2の方法における一例のチャプタ境界を示す。この第2の方法では、チャプタ境界がGOP境界と一致しているので、上述した第1の方法の場合のような不具合は、発生しない。

【 0 1 7 0 】

図27は、第1および第2の方法におけるデコード区間の例を示す。プレイリストマークPLM#1およびPLM#2の間の区間が表示すべき区間である。図27Aは、第1の方法の場合の一例のデコード区間を示す。チャプタ先頭では、エントリPTSEPStartで示される位置にプレイリストマークが設けられているので、デコード開始位置は、このエントリPTSEPStartで示される位置に対応するピクチャが含まれるGOPの先頭からとなる。すなわち、先頭側において、GOP先頭からプレイリストマークPLM#1までの区間が、表示すべき区間に対して余分にデコードする必要がある区間となる。再生時には、この余分にデコードされた区間は、マスクされる。

20

【 0 1 7 1 】

一方、第2の方法の場合は、図27Bに一例が示されるように、プレイリストマークが設けられた位置の直前のGOPからデコードを行う必要がある。これは、GOPがオープンGOPおよびクローズドGOPの何れであるかを判別できない場合、そのGOPは、オープンGOPとして、当該GOPの前のGOPの情報も用いてデコードする必要があるからである。すなわち、先頭側において、表示すべき区間に対して1GOP分、余分にデコードする区間が発生する。

30

【 0 1 7 2 】

また、時刻情報としては、エントリPTSEPStartで示される位置に対応した情報しか持っていないので、プレイリストマークは、実際には、当該プレイリストマークが設けられた位置の前後のエントリPTSEPStartで示される位置の間に存在するとししか認識できないことになる。この意味でも、プレイリストマークが設けられた位置の前のエントリPTSEPStartで示される位置のピクチャが含まれるGOPからデコードを行う必要がある。

【 0 1 7 3 】

デコード終了位置は、第1の方法においては、チャプタ終端のフレームが含まれるGOPの終端までとなる。また、第2の方法においては、上述した理由と同様にして、チャプタ終端を示すプレイリストマークの位置がその前後の何れのGOPに含まれるかを認識することができないため、当該プレイリストマークの直後に現れるエントリPTSEPStartで示される位置に対応するピクチャを含むGOPの終端までデコードを行う必要がある。このように、結果的には、第1の方法と第2の方法とで同じ位置までデコードを行う必要がある。

40

【 0 1 7 4 】

図28は、GOPがオープンGOPおよびクローズドGOPの何れかを判別可能な場合の、第1および第2の方法によるデコード区間の例を示す。プレイリストマークPLM#

50

1 および P L M # 2 の間の区間が表示すべき区間である。図 2 8 A は、プレイリストマークをエントリ PTSEPEnd で示される位置に設けた場合の例であり、上述した図 2 7 A と同等の図である。また、図 2 8 B は、プレイリストマークをオープン GOP の GOP 境界に設けた場合の例である。

【 0 1 7 5 】

既に説明したように、クローズド GOP は、その GOP 内でデコードが完結し、直前の GOP の情報を用いる必要が無い。そのため、図 2 8 C に一例が示されるように、GOP 境界にプレイリストマークを設けた場合でも、プレイリストマークを設けた位置からデコードを開始することができる。すなわち、プレイリストマークを設ける GOP がクローズド GOP であることが判別可能である場合には、先頭側において、表示すべき区間に対し
10 て余分にデコードする区間が発生しない。

【 0 1 7 6 】

また、記録が停止された後に再び記録が開始された場合、記録停止時のオープン GOP の次の GOP は、記録開始に伴いクローズド GOP となるため、記録開始位置である GOP 境界にプレイリストマークが設けられる。このため、図 2 8 B に一例が示されるように、記録の停止部分において表示すべき区間の終端とデコード区間の終端とが一致し、このプレイリストマークに基づくチャプタの終端で内容的に関連性のないフレームが表示されることがない。

【 0 1 7 7 】

なお、GOP がオープン GOP およびクローズド GOP の何れであるかを判別する手段
20 としては、幾通りかが考えられる。この発明の実施の一形態では、記録開始時は必ずクローズド GOP 構成とされるので、記録開始位置に対応する位置でチャプタ分割を行う場合には、GOP 境界にプレイリストマークを設けるようにする。

【 0 1 7 8 】

これに限らず、例えば M P E G 2 方式では、ビデオデータのエレメンタリストリーム
のヘッダに対して、GOP がクローズド GOP か否かを示すフラグが格納されるので、これを用いることが考えられる。

【 0 1 7 9 】

また、記録を行った装置と同一の装置によりチャプタ分割編集を行い、さらに同一の装置で再生を行う場合には、クローズド GOP で記録を行った位置を示す情報を装置内のメモ
30 リなどに記憶しておく方法が考えられる。さらに、装置の仕様としてクローズド GOP を用いて記録を行うようにされている場合、同一機種の装置でチャプタ分割を行う場合には、プレイリストマークを常に GOP 境界に設けるようにできる。

【 0 1 8 0 】

機種名を示す情報は、例えばインデックスファイル、ムービーオブジェクトファイル、プレイリストファイルおよびクリップインフォメーションファイルにそれぞれ設けることができる拡張データ領域（ブロック blkExtensionData() ）に格納することができる。オープン GOP およびクローズド GOP の何れを用いているかを示す情報を、これらの拡張データ領域に対して格納する方法も考えられる。

【 0 1 8 1 】

図 2 9 は、図 2 6 ~ 図 2 8 を用いて説明した各条件に関する評価を纏めて示す。図 2 9 中、「☒」（丸印）が付された項目は、有用なモードであり、「☐」（バツ印）が付された項目は、好ましくないモードである。また、「☐」（横棒）が付された項目は、実質的に意味をなさないモードである。

【 0 1 8 2 】

この図 2 9 では、チャプタ分割を行うように指定する位置が記録開始または停止位置に対応する位置であるか否かについて、それぞれ、記録と再生とにおけるチャプタ境界のずれの有無と、チャプタ先頭部および後端部のそれぞれで読み出しとデコードとが表示フレームに対してより多く必要な区間とで、プレイリストマークをエントリ PTSEPEnd の位置と、GOP 境界位置とに設けた場合について評価している。また、GOP 境界位置にプレ
50

イリストマークを設ける場合において、記録、編集および再生を同一の装置で行い装置独自の情報を利用できる場合（独自情報利用可能機とする）と、記録および編集を行った装置と再生を行う装置とが異なる場合（一般再生機とする）とのそれぞれについて評価している。

【0183】

なお、図29中で、M値は、間にBピクチャがある場合の、基準ピクチャから次の基準ピクチャまでのピクチャの移動数、N値は、1GOP内のピクチャ数を示す。例えば、GOPが「I₂B₀B₁P₅B₃B₄P₈B₆B₇P₁₁B₉B₁₀P₁₄B₁₂B₁₃」の15枚のピクチャで構成される場合、M = 3、N = 15である。

【0184】

記録開始または停止の境界位置に対応する位置でチャプタ分割を行うように指示した場合について説明する。このとき、記録と再生におけるチャプタ位置のずれに関しては、図26を用いて説明したように、プレイリストマークをエントリPTSEPStartの位置に設ける場合にずれが生じ、GOP境界位置に設ける場合には、ずれが生じない。

【0185】

デコード区間は、デコード開始位置に関しては、図28を用いて説明したように、プレイリストマークをエントリPTSEPStartの位置に設ける場合には、(M - 1)フレーム分が表示フレームに対して多く必要となり、GOP境界位置に設ける場合には、記録開始位置は必ずクローズドGOPとなるため、一般再生機および独自情報利用可能機の何れにおいても、表示フレーム区間とデコード区間とが一致する。また、デコード終了位置に関しては、プレイリストマークをエントリPTSEPStartの位置に設ける場合には、(N - M + 1)フレーム分が表示フレームに対して必要となり、GOP境界位置に設ける場合には、一般再生機および独自情報利用可能機の何れにおいても、表示フレーム区間とデコード区間とが一致する。

【0186】

記録開始または停止の境界位置ではない位置でチャプタ分割を行うように指示した場合について説明する。この場合、記録と再生におけるチャプタ位置のずれに関する評価は意味をなさないので、図29においては「-」（横棒）で示してある。

【0187】

デコード区間は、図28を用いて説明したように、デコード開始位置に関しては、プレイリストマークをエントリPTSEPStartの位置に設ける場合には、(M - 1)フレーム分が表示フレームに対して多く必要となる。一方、GOP境界位置にプレイリストマークを設ける場合には、一般再生機においては直前の1GOP分すなわちNフレーム分が表示フレームに対して必要となり、独自情報利用可能機においては、プレイリストマークが設けられるGOPがオープンGOPであると判別されれば1GOP分すなわちNフレーム分が表示フレームに対して必要になる。また、独自情報利用可能機では、当該GOPがクローズドGOPであると判別できれば、表示開始フレームとデコード開始フレームとが一致する。

【0188】

さらに、デコード終了位置に関しては、プレイリストマークをエントリPTSEPStartの位置に設ける場合には、(N - M + 1)フレーム分が表示フレームに対して多く必要となる。一方、GOP境界位置にプレイリストマークを設ける場合には、一般再生機においては直後の1GOP分すなわちNフレーム分が表示フレームに対して必要となり、独自情報利用可能機においては、表示終了フレームとデコード終了位置とが一致する。

【0189】

このように、記録開始または停止の境界位置でチャプタ分割を行う場合には、GOP境界にプレイリストマークを設け、記録開始または停止の境界位置以外でチャプタ分割を行う場合には、エントリPTSEPStartに示される位置にプレイリストマークを設けるようにすることで、ユーザの利便性を最大にできることが分かる。

【0190】

図30は、チャプタ分割編集を行う際の、プレイリストマークの一例の挿入方法を示すフローチャートである。例えばプレイリストに従いクリップを再生中にチャプタ分割が指示されると、まず、最初のステップS60で、現在再生中の位置に対応するGOPがクローズドGOPであると判定可能か否かが判断される。現在再生中の位置は、例えばビデオストリームをデコードするデコーダから取得することができる。

【0191】

若し、判定できないと判断されたら、処理はステップS61に移行し、プレイリストマークを設ける現在時刻としてエントリPTSEPStartの値を用いるものとされる。一方、ステップS60で判定可能であると判断されれば、処理はステップS63に移行され、プレイリストマークを設ける現在時刻として、GOP境界の値を用いるものとされる。

10

【0192】

ステップS61またはステップS63の処理の後、処理はステップS62に移行され、ステップS61またはステップS63で決定された現在時刻に基づきプレイリストマークを設ける処理がなされる。すなわち、このステップS62において、上述の図23においてステップS54で説明した、ブロックblkPlayListMark()内のフィールドMarkTimeStampの値が、ステップS61またはステップS63の結果に応じて決定される。

【0193】

なお、GOP境界の位置は、例えばビデオストリームをデコードするデコーダから取得することができる。これに限らず、例えば上述したエントリPTSEPStartの値に基づき求めるようにもできる。この場合には、エントリPTSEPStartの位置からGOPの構造に応じて所定フレーム分遡った位置を、GOPの先頭位置とする。

20

【0194】

記録機が記録開始操作に応じて、記録開始位置に自動的にプレイリストマークを設ける構成も考えられる。このような構成とすることで、再生時に、クリップ分割編集処理を行わなくても記録開始および停止の単位でチャプタジャンプを行うことが可能とされる。

【0195】

図31は、このような、記録開始に応じて記録開始位置に自動的にプレイリストマークを設ける場合の一例の処理を示すフローチャートである。例えばユーザにより記録機に対して記録開始操作がなされると、クリップの記録が開始されると共に、プレイリストマークを設ける現在時刻として、GOP境界の値を用いるように決定される。そして、次のステップS71で、ステップS70の結果に基づき、プレイリストマークを設ける処理が行われる。すなわち、このステップS71において、記録が開始され最初に生成されたGOPの先頭に対応する再生時刻がブロックblkPlayListMark()内のフィールドMarkTimeStampの値とされ、プレイリストマークが設けられる。

30

【0196】

なお、記録開始位置すなわち記録が開始され最初に生成されたGOPの先頭に対応する時刻は、例えばプレイリストに対して最初に記録されるクリップに関しては、値"0"に所定のオフセットを加えた値とされる。更なる記録開始および停止により生成されたクリップをプレイリストに追記するような仕様の場合には、例えば既に記録されたクリップの累積的な再生時刻に所定のオフセットを加えた値とされる。オフセットは、例えば、デコーダにストリームの供給が開始されてから当該GOPの先頭のフレームが表示される時間に対応する値であって、装置の仕様などに応じて固有の値とされる。

40

【0197】

これは、例えば1のクリップが複数回の記録開始および停止操作を含めることができるような場合に、クリップAVストリームファイル、クリップインフォメーションファイルおよび当該クリップインフォメーションファイルに対応するプレイアイテムが1つのまま、プレイリストマークのみで、記録開始および停止操作に対応するチャプタ境界を設ける場合に用いて好適な方法である。

【0198】

なお、1のプレイリストファイルに設けることができるプレイリストマークの数に対し

50

て、上限を設定することができる。プレイリストマーク数の上限を設定することで、無制限にプレイリストファイルの容量が増大するのが防がれる。また、例えば編集機や再生機におけるチャプタ表示数に、例えば3桁などの制限が設けられているような場合に、プレイリストマークの上限を設定することとも考えられる。上述した図30および図31を用いて説明した処理は、それぞれ、現在設定しようとするプレイリストマークを設定することでこのプレイリスト内に設定可能なプレイリストマーク数の上限を超えないか否かの判断を行った上で、上限を超えないとの判断がなされた後に、行われる。

【0199】

次に、この発明の実施の一形態に適用可能な一例の記録再生装置について説明する。先ず、記録再生装置の説明に先んじて、仮想プレーヤについて、概略的に説明する。上述したようなデータ構造を有するディスクがプレーヤに装填されると、プレーヤは、ディスクから読み出されたムービーオブジェクトなどに記述されたコマンドを、プレーヤ内部のハードウェアを制御するための固有のコマンドに変換する必要がある。プレーヤは、このような変換を行うためのソフトウェアを、プレーヤに内蔵されるROM(Read Only Memory)にあらかじめ記憶している。このソフトウェアは、ディスクとプレーヤを仲介してプレーヤにこのフォーマットの規定に従った動作をさせることから、仮想プレーヤと称される。

【0200】

図32は、この仮想プレーヤの動作を概略的に示す。図32Aは、ディスクのローディング時の動作の例を示す。ディスクがプレーヤに装填されディスクに対するイニシャルアクセスがなされると(ステップS30)、1のディスクにおいて共有的に用いられる共有パラメータが記憶されるレジスタが初期化される(ステップS31)。そして、次のステップS32で、ムービーオブジェクトなどに記述されたプログラムがディスクから読み込まれて実行される。なお、イニシャルアクセスは、ディスク装填時のように、ディスクの再生が初めて行われることをいう。

【0201】

図32Bは、プレーヤが停止状態からユーザにより例えばプレイキーが押下され再生が指示された場合の動作の例を示す。最初の停止状態(ステップS40)に対して、ユーザにより、例えばリモートコントロールコマンドなどを用いて再生が指示される(UO: User Operation)。再生が指示されると、先ず、レジスタすなわち共通パラメータが初期化され(ステップS41)、次のステップS42で、ムービーオブジェクト実行フェーズに移行する。

【0202】

ムービーオブジェクトの実行フェーズにおけるプレイリストの再生について、図33を用いて説明する。UOなどにより、タイトル番号#1のコンテンツを再生開始する指示があった場合について考える。プレーヤは、コンテンツの再生開始指示に応じて、上述した図2に示されるインデックステーブル(Index Table)を参照し、タイトル#1のコンテンツ再生に対応するオブジェクトの番号を取得する。例えばタイトル#1のコンテンツ再生を実現するオブジェクトの番号が#1であったとすると、プレーヤは、ムービーオブジェクト#1の実行を開始する。

【0203】

この図33の例では、ムービーオブジェクト#1に記述されたプログラムは2行からなり、1行目のコマンドが"Play Playlist(1)"であるとすると、プレーヤは、プレイリスト#1の再生を開始する。プレイリスト#1は、1以上のプレイアイテムから構成され、プレイアイテムが順次再生される。プレイリスト#1中のプレイアイテムの再生が終了すると、ムービーオブジェクト#1の実行に戻り、2行目のコマンドが実行される。図33の例では、2行目のコマンドが"jump MenuTitle"であって、このコマンドが実行されインデックステーブルに記述されたメニュータイトル(MenuTitle)を実現するムービーオブジェクトの実行が開始される。

【0204】

図34は、この発明の実施の一形態に適用可能な記録再生装置の一例の構成を概略的に

10

20

30

40

50

示す。この図 3 4 に例示される記録装置は、外部から入力されるビデオデータおよびオーディオデータを記録媒体に記録し、記録媒体に記録されたビデオデータおよびオーディオデータを再生する、単独の記録再生装置として用いることもできるし、光学系や撮像素子などを備えたカメラブロックと組み合わせ、撮像した撮像信号に基づくビデオデータを記録媒体に記録する、ビデオカメラ装置の記録ブロックとして用いることもできる。

【 0 2 0 5 】

適用可能な圧縮符号化や多重化の方式としては、様々に考えられる。例えば、H . 2 6 4 | A V C に規定される方式を、この発明の実施の一形態の圧縮符号化として適用することができる。これに限らず、M P E G 2 方式に基づき圧縮符号化を行うようにしてもよい。また、多重化方式は、例えば M P E G 2 システムズを適用することができる。以下では、ビデオデータの圧縮符号化を H . 2 6 4 | A V C に規定される方式に準じて行い、ビデオデータおよびオーディオデータの多重化を、M P E G 2 システムズに規定される方式に準じて行うものとして説明する。

【 0 2 0 6 】

制御部 3 0 は、例えば C P U (Central Processing Unit)、R A M (Random Access Memory) および R O M (Read Only Memory) などからなり (図示しない)、R O M に予め記憶されたプログラムやデータに基づき、R A M をワークメモリとして用いてこの記録装置の記録部 1 0 の各部を制御する。なお、制御部 1 0 と記録部 1 0 の各部とを接続する経路は、複雑さを避けるために、図 3 4 では省略している。

【 0 2 0 7 】

U I (User Interface) 部 3 1 は、この記録装置の動作をユーザが操作するための操作子が所定に設けられ、操作子に対する操作に応じた制御信号を出力する。この制御信号は、制御部 3 0 に供給される。制御部 3 0 は、ユーザ操作に応じて U I 部 3 1 から供給された制御信号に基づきなされるプログラムの処理により、記録再生部 1 0 の各部の動作を制御する。また、U I 部 3 1 は、簡易的な表示部を有し、後述する管理情報処理部 1 6 から供給される情報に基づき、所定の表示、例えば記録再生装置における記録や再生の状態を示す表示を行うことができるようにされている。

【 0 2 0 8 】

例えば、U I 部 3 1 に対してなされた操作に応じて、記録再生装置による記録媒体 3 2 に対してデータを記録する動作の開始および停止の動作や、記録媒体 3 2 からデータを再生する再生動作が制御部 3 0 により制御される。また例えば、編集動作時には、U I 部 3 1 に対してなされた操作に応じて記録媒体 3 2 からデータが再生されると共に、操作に応じた位置でのチャプタ分割処理がなされる。

【 0 2 0 9 】

ビデオエンコーダ 1 1 は、複数フレームのビデオデータを格納可能なバッファメモリを有し、供給されたベースバンドのデジタルビデオデータをバッファメモリに溜め込んで、所定の方式で圧縮符号化する。H . 2 6 4 | A V C に規定される方式に準じて圧縮符号化がなされるこの例では、例えば、D C T (Discrete Cosine Transform) と画面内予測とによりフレーム内圧縮を行うと共に、動きベクトルを用いたフレーム間圧縮を行い、さらにエントリピー符号化を行い圧縮効率を高める。ビデオエンコーダ 1 1 で圧縮符号化されたデジタルビデオデータは、H . 2 6 4 | A V C エレメンタリストリーム (E S) として出力される。

【 0 2 1 0 】

ビデオデコーダ 2 0 は、複数フレームのビデオデータを格納可能なバッファメモリを有し、供給された圧縮ビデオデータをバッファメモリに溜め込んで、圧縮符号化方式に対応した復号化方式でデコードし、ベースバンドのデジタルビデオデータとして出力する。例えば、ビデオエンコーダ 1 1 が H . 2 6 4 | A V C に規定される方式に準じて圧縮符号化を行うこの例では、ビデオデコーダ 2 0 もビデオエンコーダ 1 1 に対応して、H . 2 6 4 | A V C に規定される方式に準じてデコード処理を行う。ビデオデコーダ 2 0 は、後述するマルチプレクサ / デマルチプレクサ 1 3 (以下、M U X / D E M U X 1 3) で抽出さ

10

20

30

40

50

れる、D T S (Decoding Time Stamp)およびP T S (Presentation Time Stamp)で示される時刻に基づき、デコードおよび出力を行うことができる。ビデオデコーダ20でデコードされて得られたベースバンドのデジタルビデオデータは、端子42から出力される。

【0211】

オーディオエンコーダ12は、端子41から供給されたベースバンドのデジタルオーディオデータを所定の圧縮符号化方式、例えばドルビーデジタル(Dolby Digital：登録商標)方式により圧縮符号化する。オーディオデータの圧縮符号化方式は、これに限られるものではなく、さらに、オーディオデータを圧縮符号化せず、ベースバンドのデータのまま用いることも考えられる。

【0212】

オーディオデコーダ21は、供給された圧縮オーディオデータを圧縮符号化方式に対応した復号化方式でデコードし、ベースバンドのデジタルオーディオデータとして出力する。オーディオエンコーダ12がドルビーデジタル方式により圧縮符号化されるこの例では、オーディオデコーダ21もオーディオエンコーダ12に対応して、ドルビーデジタル方式に準じた復号化方式でデコードがなされる。デコードされたオーディオデータは、ビデオデコーダ20から出力されるビデオデータと同期的に、端子43から出力される。

【0213】

M U X / D E M U X 13は、それぞれ圧縮符号化されて供給されたデジタルビデオデータおよびデジタルオーディオデータを所定の方式で多重化し、1本のデータストリームとして出力するマルチプレクサ機能と、デジタルビデオデータとデジタルオーディオデータとが所定に多重化されたデータストリームから、デジタルビデオデータとデジタルオーディオデータとを分離してそれぞれ取り出す、デマルチプレクサ機能とを有する。

【0214】

マルチプレクサ機能は、例えば、M P E G 2システムズに準じて多重化が行われるこの例では、M P E G 2のトランスポートストリームを用いて、供給された圧縮ビデオデータおよび圧縮オーディオデータを時分割で多重化する。例えば、M U X / D E M U X 13は、バッファメモリを有し、供給された圧縮ビデオデータおよび圧縮オーディオデータを一旦バッファメモリに格納する。バッファメモリに格納された圧縮ビデオデータは、所定サイズ毎に分割されヘッダが付加されて、P E S (Packetized Elementary Stream)パケット化される。圧縮オーディオデータも同様に、所定サイズ毎に分割されヘッダが付加されてP E Sパケット化される。ヘッダには、パケットに格納されるデータの再生時刻を示すP T Sや復号時刻を示すD T S (Decoding Time Stamp)といった、M P E G 2システムズに規定される所定の情報が格納される。P E Sパケットは、さらに分割されてトランスポートパケット(T Sパケット)のペイロードに詰め込まれる。T Sパケットのヘッダには、ペイロードに詰め込まれたデータ種別などを識別するためのP I D (Packet Identification)が格納される。

【0215】

デマルチプレクサ機能は、マルチプレクサ機能と逆の処理を行い、パケットから圧縮ビデオデータおよび圧縮オーディオデータを抽出する。例えば、供給されたT SパケットのヘッダからP I Dを検出し、T Sパケットをペイロードに格納されるデータ種別毎に振り分ける。そして、振り分けられたT Sパケットのそれぞれについて、ペイロードに格納されたデータを取り出し、P E Sパケットを再構築する。さらに、P E Sパケットのペイロードに格納された圧縮ビデオデータや圧縮オーディオデータを取り出し、P E Sヘッダに格納された情報などに基づきヘッダ情報などを付加し、それぞれ1本のエレメンタリストリームとして出力する。

【0216】

ストリームバッファ14は、M U X / D E M U X 13(記録時)または記録再生制御部15(再生時)から供給されたT Sパケットを一時的に格納する。ストリームバッファ1

10

20

30

40

50

4 に対する T S パケットの読み書きのタイミングを所定に制御することで、記録媒体 3 2 に対するアクセス速度と、オーディオおよびビデオデータエンコードやデコードなどの信号処理速度との間の整合性をとる。

【 0 2 1 7 】

記録再生制御部 1 5 は、記録媒体 3 2 に対するデータの記録と、記録媒体 3 2 からのデータの再生を制御する。すなわち、記録再生制御部 1 5 は、例えば制御部 3 0 といった上位からの命令に基づき、指定されたアドレスに対するデータの書き込みや、指定されたアドレスからのデータの読み出しを行う。

【 0 2 1 8 】

記録媒体 3 2 としては、例えば記録可能なタイプの D V D (Digital Versatile Disc) を用いることができる。これに限らず、記録媒体 3 2 としてハードディスクドライブを用いてもよいし、半導体メモリを記録媒体 3 2 に適用することも可能である。また、記録媒体 3 2 として、より大容量を実現した B l u - r a y D i s c (ブルーレイディスク：登録商標) を適用することも考えられる。

【 0 2 1 9 】

管理情報処理部 1 6 は、例えば C P U、ワークメモリとしての R A M およびプログラム所定のデータが予め記憶される R O M からなる (図示しない)。これに限らず、管理情報処理部 1 6 は、例えば制御部 3 0 におけるプログラム処理で管理情報処理部 1 6 の機能を実現することも可能である。この場合、例えば制御部 3 0 の有する R A M が揮発性メモリ 1 7 として用いられると共に、不揮発性メモリ 1 8 が制御部 3 0 に接続される。

【 0 2 2 0 】

管理情報処理部 1 6 は、例えば揮発性メモリ 1 7 をワークメモリとして用いて、上述したインデックスファイル "index.bdmv"、ムービーオブジェクトファイル "MovieObject.bdmv"、プレイリストファイル "xxxxx.mpls" およびクリップインフォメーションファイル "zzzz.clpi" に関する処理を行う。例えばこの発明の実施の一形態によるプレイリストマークの追加挿入処理は、制御部 3 0 の制御に基づきこの管理情報処理部 1 6 により行われる。

【 0 2 2 1 】

このような構成を有する記録再生装置における、記録時の動作について説明する。なお、以下では、ビデオデータは、H . 2 6 4 | A V C に準じた方式で圧縮符号化され、オーディオデータは、A A C 方式で圧縮符号化されるものとする。

【 0 2 2 2 】

ベースバンドのデジタルビデオデータが端子 4 0 から記録再生部 1 0 に入力され、ビデオエンコーダ 1 1 に供給される。例えば、U I 部 3 1 に対して記録開始の操作がなされると、ビデオエンコーダ 1 1 は、供給されたデジタルビデオデータの圧縮符号化を開始する。ビデオエンコーダ 1 1 は、ベースバンドのデジタルビデオデータを圧縮符号化して H . 2 6 4 | A V C のエレメンタリストリーム (E S) として出力する。このエレメンタリストリームは、M U X / D E M U X 1 3 に供給される。

【 0 2 2 3 】

ベースバンドのデジタルオーディオデータが端子 4 1 から記録再生部 1 0 に入力され、オーディオエンコーダ 1 2 に供給される。オーディオエンコーダ 1 2 は、上述の U I 部 3 1 に対する記録開始の操作に応じたタイミングで供給されたオーディオデータの圧縮符号化を開始する。オーディオエンコーダ 1 2 で圧縮符号化されたデジタルオーディオデータは、M U X / D E M U X 1 3 に供給される。

【 0 2 2 4 】

M U X / D E M U X 1 3 は、それぞれ圧縮符号化されて供給されたデジタルビデオデータおよびデジタルオーディオデータを所定の方式で多重化し、1 本のデータストリームとして出力する。例えば、M U X / D E M U X 1 3 は、バッファメモリを有し、供給された圧縮ビデオデータおよび圧縮オーディオデータを一旦バッファメモリに格納する。

【 0 2 2 5 】

バッファメモリに格納された圧縮ビデオデータは、所定構造毎に分割されヘッダが付加

10

20

30

40

50

されて、P E S (Packetized Elementary Stream) パケット化される。圧縮オーディオデータも同様に、所定サイズ毎に分割されヘッダが付加されて P E S パケット化される。ヘッダには、パケットに格納されるデータの再生時刻を示す P T S や復号時刻を示す D T S (Decoding Time Stamp) といった、M P E G 2 システムズに規定される所定の情報が格納される。P E S パケットは、さらに分割されてトランスポートパケット (T S パケット) のペイロードに詰め込まれる。T S パケットのヘッダには、ペイロードに詰め込まれたデータを識別するための P I D (Packet Identification) が格納される。マルチプレクサ 1 3 から出力された T S パケットは、ストリームバッファ 1 4 に一旦溜め込まれる。

【 0 2 2 6 】

なお、T S パケットは、実際には、M U X / D E M U X 1 3 においてさらに所定サイズのヘッダが付加されて出力される。この、T S パケットに対して所定のヘッダを付加したパケットを、ソースパケットと呼ぶ。

【 0 2 2 7 】

記録制御部 1 5 は、ストリームバッファ 1 4 に溜め込まれたデータ量を監視し、ストリームバッファ 1 4 に所定量以上のデータが溜め込まれると、ストリームバッファ 1 4 から記録媒体 3 2 の記録単位毎にデータを読み出して記録媒体 3 2 に書き込む。

【 0 2 2 8 】

管理情報処理部 1 6 は、記録データに基づき、揮発性メモリ 1 7 をワークメモリとして用いて、上述したインデックスファイル "index.bdmv"、ムービーオブジェクトファイル "MovieObject.bdmv"、プレイリストファイル "xxxxx.mpls" およびクリップインフォメーションファイル "zzzzz.clpi" に格納するための情報を生成する。生成された情報は、所定のタイミングで記録媒体 3 2 に書き込まれる。

【 0 2 2 9 】

一例として、管理情報処理部 1 6 は、M U X / D E M U X 1 3 から記録データの時刻情報を取得すると共に、記録制御部 1 5 から記録データの記録媒体 3 2 におけるアドレス情報を取得し、取得されたこれら時刻情報およびアドレス情報に基づきエントリ PTSEStart が生成される。また、U I 部 3 1 に対する記録開始、記録終了の操作に応じて制御部 3 0 から出力される制御信号と、M U X / D E M U X 1 3 および記録制御部 1 5 からの記録データに関する情報とに基づき、プレイリストファイル "xxxxx.mpls" の生成または更新、クリップインフォメーションファイル "zzzzz.clpi" の生成などが行われる。

【 0 2 3 0 】

この記録再生装置が記録開始に伴い記録開始位置にプレイリストマークを設けるようにされている場合、U I 部 3 1 に対する記録開始操作に応じて生成される先頭のフレームに対応する再生時刻をフィールド MarkTimeStamp の値として持つプレイリストマークが、管理情報処理部 1 6 において作成され、プレイリストに対して追加される。

【 0 2 3 1 】

再生時の動作について説明する。記録媒体 3 2 が例えば記録再生装置に対して装填されると、記録媒体 3 2 から、インデックスファイル "index.bdmv"、ムービーオブジェクトファイル "MovieObject.bdmv" が読み込まれ、管理情報処理部 1 6 に渡される。管理情報処理部 1 6 は、これらの情報を、揮発性メモリ 1 7 に記憶する。制御部 3 0 は、管理情報処理部 1 6 から揮発性メモリ 1 7 に記憶された情報を取得し、取得された情報に基づき、例えば、U I 部 3 1 が有する表示部に対して記録媒体 3 2 に記録されているクリップに関する情報を表示させる。ユーザは、この情報に基づき U I 部 3 1 に対して所定の操作を行うことで、記録媒体 3 2 に記録されているクリップの再生を指示することができる。なお、ここでは、簡単のために、クリップの先頭から再生を開始させるものとする。

【 0 2 3 2 】

制御部 3 0 は、U I 部 3 1 に対する操作に応じて、管理情報処理部 1 6 からプレイリストファイルの情報を取得し、このプレイリストファイルに基づき、記録再生制御部 1 5 に対して、当該プレイリストファイルに格納されるプレイアイテムに参照されるクリップインフォメーションファイルと、対応するクリップ A V ストリームファイルを記録媒体 3 2

10

20

30

40

50

から読み出すように命令を出す。記録再生制御部 15 は、この命令に従い、記録媒体 32 からクリップインフォメーションファイルとクリップ A V ストリームファイルとを読み出す。クリップ A V ストリームファイルは、記録媒体 32 から T S パケット毎に読み出され、記録再生制御部 15 を介してストリームバッファ 14 に溜め込まれる。

【0233】

なお、実際には、記録媒体 32 から読み出される T S パケットは、本来の T S パケットに対して所定サイズのヘッダが付加されたソースパケットである。

【0234】

M U X / D E M U X 13 は、ストリームバッファ 14 に溜め込まれたデータ量を監視し、ストリームバッファ 14 に所定量以上の T S パケットが溜め込まれると、ストリームバッファ 14 からビデオデコーダ 20 がデコードに必要とする分のデータを、T S パケット単位で読み出す。読み出された T S パケットは、M U X / D E M U X 13 に供給されバッファメモリに一旦格納され、P I D に基づきデータ種類毎に振り分けられペイロードに格納されたデータから P E S パケットが再構築される。さらに、P E S パケットのペイロードからデータが取り出されると共に、P E S ヘッダの情報に基づき、D T S や P T S といったデコードや再生の時刻を指定する情報など、所定にヘッダ情報などが付加され、圧縮符号化されたビデオデータおよびオーディオデータのエレメンタリストリームがそれぞれ生成される。

【0235】

圧縮ビデオデータは、ビデオデコーダ 20 に供給される。ビデオデコーダ 20 は、供給された圧縮ビデオデータをバッファメモリに溜め込み、所定ピクチャ分のデータが溜め込まれたら、バッファメモリに溜め込まれたデータに対してデコード処理を開始する。デコードされたビデオデータは、例えば図示されないシステムクロックから供給される S T C (System Time Clock) に基づき、P T S に従いフレームタイミングで順次出力される。

【0236】

一方、圧縮オーディオデータは、オーディオデコーダ 21 に供給され、所定にデコード処理がなされる。デコードされたオーディオデータは、ビデオデコーダ 20 から出力されるビデオデータと同期的に、オーディオデコーダ 21 から出力される。

【0237】

この記録再生装置は、記録媒体 32 に記録されたクリップに対するチャプタ分割編集を行うことができる。上述のようにして記録媒体 32 からクリップを再生中に、U I 部 31 に対して所定の操作を行い、チャプタ分割を行う位置を指定する。チャプタ分割の位置を指定する方法としては、クリップを通常通り、例えば順方向の 1 倍速再生で再生しながら所望の位置でクリップ分割の操作を行う方法や、高速再生、スロー再生、コマ送り再生といった可変速再生を用いて分割位置をフレーム単位で検索してチャプタ分割の操作を行う方法などが考えられる。

【0238】

分割位置が指定されると、図 23 を用いて説明した処理に従い、指定された分割位置に対応する現在の再生時刻と、プレイリスト中の他のプレイリストマークとの位置関係が判断される。それと共に、図 30 を用いて説明した処理に従い、プレイリストマークを設ける値として、エントリ P T S E P S t a r t および G O P 境界の何れの値を用いるかを判断する。

【0239】

なお、プレイリストマークを設ける現在時刻としてエントリ P T S E P S t a r t の値を用いる場合で、分割位置をフレーム単位で任意に指定するとき、指定された分割位置が必ずしもエントリ P T S E P S t a r t で示されるフレームであるとは限らない。この場合、例えば指定された位置に最も近いエントリ P T S E P S t a r t の値を、指定された分割位置に対応する P T S E P S t a r t の値として用いることが考えられる。これに限らず、指定された位置の直前または直後のエントリ P T S E P S t a r t の値を用いるようにしてもよい。

【0240】

また、現在再生中の位置に対応する G O P がクローズド G O P であると判断できた場合

10

20

30

40

50

は、クローズドGOPの先頭のフレームの再生時刻は、例えばMUX/DEMUX13から取得することができる。これに限らず、エントリPTSEPStartの値を基点とし、ビデオコーデック20のバッファメモリに溜め込まれているデータとSTCとに基づき、クローズドGOPの先頭のフレームの再生時刻を求めることも考えられる。

【0241】

図35は、上述のようにして設けられたプレイリストマークに基づくチャプタジャンプ処理の一例の方法を示すフローチャートである。一例として、上述のようにしてプレイリストマークが設けられチャプタ分割編集されたプレイリストを再生中に、例えばUI部31に対してチャプタジャンプを行うような操作がなされる。制御部30は、この操作に応じて、まず、現在再生中の位置に対応する時刻情報を求める(ステップS80)。例えば、制御部30は、MUX/DEMUX13との間でやりとりを行い、MUX/DEMUX13で分離されるビデオデータのエレメンタリストリームに含まれるPTSと、MUX/DEMUX13におけるSTCとに基づき、ビデオコーデック20でデコード中の任意のフレームの再生時刻を知ることができる。

【0242】

なお、現在再生中の位置に対応する時刻情報を求める方法は、これに限られない。例えば、再生中、常にクリップインフォメーションファイル中のブロックblkEMap()(図17参照)の情報をうい、エントリPTSEPStartに基づきアドレス変換を行う方法が考えられる。再生中には、次のGOPのデータを読み出す必要があり、この方法によれば、次のGOPのデータを容易に取得することが可能となる。また、現在の再生位置と、プレイリストマークの位置との関係を常に更新しながら再生を行うことで、UI部31の表示画面に対して現在再生中のチャプタの情報を表示することが可能となり、好ましい。

【0243】

現在再生中の位置に対応する時刻情報が求められたら、次のステップS81で、制御部30により、当該時刻情報で示される時刻に対して再生方向に向けて最も近くにあるプレイリストマークが、再生中のプレイリストのブロックblkPlayListMark()から検索される。

【0244】

該当するプレイリストマークが存在するとされれば(ステップS82)、処理はステップS83に移行され、制御部30により、ステップS81で検索されたプレイリストマークに基づき、当該プレイリストマークに対応するエントリPTSEPStartが検索される。

【0245】

この実施の一形態では、プレイリストマークは、エントリPTSEPStartの位置またはGOP境界位置の何れかに設けられる。したがって、プレイリストマークに対応するエントリPTSEPStartは、プレイリストマークの再生時刻を示す値MarkTimeStampと値が一致するエントリPTSEPStartが存在すれば、そのエントリPTSEPStartとされる。一方、プレイリストマークの再生時刻を示す値MarkTimeStampと値が一致するエントリPTSEPStartが存在しない場合に、プレイリストマークがGOP境界に設けられ且つGOPがクローズドGOPであることが分かる場合には、値MarkTimeStampから再生時刻を再生方向に進めていったときに最初に現れるエントリPTSEPStartとされる。さらに、プレイリストマークの再生時刻を示す値MarkTimeStampと値が一致するエントリPTSEPStartが存在しない場合に、プレイリストマークがGOP境界に設けられていないか、または、GOPがクローズドGOPであることが分からない場合は、再生方向とは逆方向に進めて次に現れるエントリPTSEPStartとされる。

【0246】

次のステップS84で、制御部30により、再生中のプレイアイテムに対応するクリップインフォメーションファイル内のブロックblkEMap()(図17～図20参照)に基づき、ステップS83で検索されたエントリPTSEPStartに対応するアドレス情報が求められる。そして、制御部30から記録再生制御部15に対して、記録媒体32から当該アドレス情報で示されるアドレスからの読み出しを行うように命令される。

【 0 2 4 7 】

記録媒体 3 2 から読み出されたデータは、記録再生制御部 1 5 からストリームバッファ 1 4 を介して M U X / D E M U X 1 3 に供給され、パケットを用いて時分割多重されたストリームからデータ種類毎の分離がなされる。分離されたデータうち圧縮ビデオデータのエレメンタリストリームは、ビデオデコーダ 2 0 に供給される。ビデオデコーダ 2 0 は、供給された圧縮ビデオデータのエレメンタリストリームのデコードを開始する（ステップ S 8 5）。そして、プレイリストマークで示されるフレームがデコードされ出力可能になったら、そのフレームからの出力がなされ、表示が開始される（ステップ S 8 6）。

【 0 2 4 8 】

このとき、プレイリストマークがエントリ PTSEPStart 位置に設けられている場合には、G O P において、当該エントリ PTSEPStart に対応するピクチャより前のピクチャは、デコードされるが表示は行わないようにされる。一方、プレイリストマークが G O P 境界位置に設けられている場合には、G O P の先頭位置から表示が開始される。

10

【 0 2 4 9 】

なお、再生時刻で最後に設けられたプレイリストマークよりも後ろを再生中にチャプタジャンプを指示されることも考えられる。この場合、現在再生中の位置に対して再生方向に向けて最も近いプレイリストマークが、再生中のプレイリスト中に存在しないことになる。この場合、チャプタジャンプの指示が無視される。これに限らず、例えば図 3 5 のステップ S 8 7 に一例が示されるように、当該プレイリストの終端までジャンプするようにしてもよい。

20

【 0 2 5 0 】

次に、この発明の実施の一形態の他の例について説明する。上述では、この発明が単体の記録装置に適用された例について説明した（図 3 4 参照）。これに対し、この実施の一形態の他の例では、この発明を、撮像素子と、被写体からの光を撮像素子に入射させる光学系とを有し、撮像素子で撮像された撮像信号に基づきビデオデータを記録媒体に記録するようにした、ビデオカメラ装置に適用した。

【 0 2 5 1 】

図 3 6 は、この発明の実施の一形態の他の例によるビデオカメラ装置 1 0 0 の一例の構成を示す。ビデオカメラ装置 1 0 0 において、記録系の構成は、図 3 4 を用いて説明した記録装置の構成を略そのまま適用できるので、図 3 4 と共通する部分には同一の符号を付し、詳細な説明を省略する。

30

【 0 2 5 2 】

図 3 6 の構成において、カメラ部 5 0 は、映像信号に関する構成として、光学系 5 1、撮像素子 5 2、撮像信号処理部 5 3、カメラ制御部 5 4、ビデオ信号処理部 5 8 および表示部 5 5 を有し、音声信号に関する構成として、マイクロフォン（M I C）5 6、音声信号処理部 5 7 およびスピーカ部 S P 6 0 を有する。制御部 3 0 は、カメラ部 5 0 の各部との間で各種制御信号や情報のやりとりを行い、カメラ部 5 0 の動作を制御する。また、制御部 5 0 は、ユーザ操作に応じて U I 部 3 1 から供給される制御信号に基づき、カメラ部 5 0 の動作を制御する。

【 0 2 5 3 】

なお、ビデオカメラ装置 1 0 0 として構成される場合、記録開始操作および記録停止操作は、例えば、U I 部 3 1 に設けられた単一の記録スイッチを用い、当該記録スイッチが押下される毎に記録開始および記録停止が交互に指示されるようになされるのが一般的である。また、このビデオカメラ装置 1 0 0 では、記録媒体 2 0 として、B l u - r a y D i s c や記録可能なタイプの D V D といった、ディスク記録媒体を適用するものとする。

40

【 0 2 5 4 】

カメラ部 5 0 において、光学系 5 1 は、被写体からの光を撮像素子 5 2 に導くためのレンズ系、絞り調整機構、フォーカス調整機構、ズーム機構、シャッタ機構などを備える。絞り調整機構、フォーカス調整機構、ズーム機構およびシャッタ機構の動作は、制御部 3

50

0 から供給される制御信号に基づき、カメラ制御部 5 4 により制御される。

【 0 2 5 5 】

撮像素子 5 2 は、例えば C C D (Charge Coupled Device) からなり、光学系 5 1 を介して照射された光を光電変換により電気信号に変換し、所定の信号処理を施し撮像信号として出力する。撮像信号処理部 5 3 は、撮像素子から出力された撮像信号に対して所定の信号処理を施し、ベースバンドのデジタルビデオデータとして出力する。例えば撮像信号処理部 5 3 は、撮像素子 5 2 から出力された撮像信号に対して、C D S (Correlated Double Sampling) 回路により画像情報を有する信号だけをサンプリングすると共に、ノイズを除去し、A G C (Auto Gain Control) 回路によりゲインを調整する。そして、A / D 変換によりデジタル信号に変換する。

10

【 0 2 5 6 】

また、撮像信号処理部 5 3 は、撮像素子 5 2 から出力された撮像信号の情報を制御部 3 0 に送る。制御部 3 0 は、この情報に基づき光学系 5 1 を制御するための制御信号を生成し、カメラ制御部 5 4 に供給する。カメラ制御部 5 4 は、この制御信号に基づきフォーカス調整機構や絞り調整機構などの制御を行う。

【 0 2 5 7 】

ビデオ信号処理部 5 8 は、供給されたデジタル信号に対して所定の信号処理を施す。例えば、ビデオ信号処理部 5 8 は、供給されたデジタル信号に対して検波系の信号処理を施し、R (赤色) 、G (緑色) および B (青色) 各色の成分を取り出す。そして、取り出された各色成分に基づき補正やホワイトバランス補正などの処理を行い、最終的に 1

20

【 0 2 5 8 】

表示部 5 5 は、例えば L C D (Liquid Crystal Display) を表示素子として用い、ビデオ信号処理部 5 8 から供給されたデジタルビデオデータに基づく表示を行うことができる。表示部 5 5 は、撮影時には撮影画像のモニタとして用いられ、再生時には、再生画像を映出させることができる。

【 0 2 5 9 】

音声信号処理部 5 7 は、例えばマイクロフォン M I C 5 6 から供給されるアナログ音声信号を A / D 変換してデジタルオーディオデータとし、ノイズ除去や音質補正など所定の音声信号処理を施してベースバンドのデジタルオーディオデータとして出力する。また、音声信号処理部 5 7 は、供給されるデジタルオーディオデータに対して音質補正や音量調整などの所定の音声信号処理を施して D / A 変換してアナログ音声信号とし、増幅処理などを行いスピーカ部 S P 6 0 に供給する。

30

【 0 2 6 0 】

撮影時には、光学系 5 1 を介して撮像素子 5 2 に入射された光が光電変換により電気信号に変換され撮像信号として出力される。撮像信号は、撮像信号処理部 5 3 で所定に信号処理され、A / D 変換されデジタルビデオ信号として出力される。このデジタルビデオ信号は、ビデオ信号処理部 5 8 に供給される。ビデオ信号処理部 5 8 は、供給されたデジタルビデオ信号に対して画質補正などの所定の信号処理を施してデジタルビデオデータとして出力する。このデジタルビデオデータは、記録再生部 1 0 に供給され端子 4 0 に入力される。

40

【 0 2 6 1 】

また、ビデオ信号処理部 5 8 は、撮像信号処理部 5 3 から供給されたデジタル信号に基づき、表示部 5 5 に表示するためのデジタルビデオデータを生成する。さらに、ビデオ信号処理部 5 8 は、制御部 3 0 とやりとりを行い、制御部 3 0 で所定に生成された表示制御信号に基づく画像を生成することができる。このデジタルビデオデータや画像は、表示部 5 5 に供給され、映出される。

【 0 2 6 2 】

一方、マイクロフォン 5 6 から出力された音声信号は、音声信号処理部 5 7 でノイズ除

50

去、リミッタ処理、音質補正などの所定の信号処理を施され、さらにA/D変換され、デジタルオーディオデータとして出力される。このデジタルオーディオデータは、記録再生部10に供給され、端子41に入力される。

【0263】

記録停止状態からUI部31に設けられた記録スイッチが押下されると、記録開始を指示する制御信号がUI部31から制御部30に供給され、制御部30の制御に基づきカメラ部50から出力されたベースバンドのデジタルビデオデータおよびデジタルオーディオデータの記録媒体32への記録が開始される。

【0264】

すなわち、図34を用いて既に説明したように、制御部30の制御に基づきビデオエンコーダ11およびオーディオエンコーダ12の動作が開始され、ビデオデータおよびオーディオデータがそれぞれビデオエンコーダ11およびオーディオエンコーダ12で圧縮符号化され、マルチプレクサ13で所定にパケット化され多重化されてAVストリームデータとされる。AVストリームデータは、ストリームバッファ14を介して、記録制御部15に供給され、クリップAVストリームファイルとして記録媒体32に記録される。

【0265】

ここで、ビデオカメラ装置100が記録開始に応じて記録開始位置に自動的にプレイリストを設けるようにされている場合、図31を用いて説明した処理に従い、記録開始位置に対応する位置に対して自動的にプレイリストマークが設けられる。実際には、プレイリストファイルに格納される情報は、例えば揮発性メモリ17上において生成される。このプレイリストマークも、この時点では、揮発性メモリ17上の情報である。

【0266】

UI部31の記録スイッチが次に押下されると、記録が停止され、クリップインフォメーションファイルの作成や、プレイリストファイルの更新が行われる。管理情報処理部16は、マルチプレクサ13および記録制御部15からの情報に基づき、記録媒体32に記録されるクリップAVストリームファイルに対応するクリップインフォメーションファイルを作成する。また、管理情報処理部16は、当該クリップインフォメーションファイルを参照するプレイアイテムを生成し、既にプレイリストが存在する場合には、生成されたプレイアイテムを当該プレイリストに対して追加すると共に、プレイリストに対してプレイリストマークの情報を格納する。

【0267】

再生時には、記録媒体32がビデオカメラ装置100に装填されると、記録媒体32に記録されているインデックスファイル、ムービーオブジェクトファイルなどのファイルが読み出され、管理情報処理部16に供給される。制御部30は、管理情報処理部16からインデックスファイルの情報を取得し、取得された情報に基づき所定のメニュー画面を表示するための表示制御信号を生成する。この表示制御信号は、ビデオ信号処理部58に供給され、表示部55に表示される。この表示に応じてUI部31に対して所定の操作を行うと、例えば、記録媒体32から再生が指示されたプレイリストファイルが読み出され、このプレイリストファイルの記述に従い、記録媒体32に記録されたクリップの再生がなされる。

【0268】

すなわち、図34を用いて既に説明したように、制御部30は、UI部31に対する操作に応じて管理情報処理部16からプレイリストファイルの情報を取得し、取得された情報に基づき、記録再生制御部15に対して記録媒体32からクリップインフォメーションファイルやクリップAVストリームファイルを読み出すように命令を出す。

【0269】

記録媒体32から読み出されたクリップAVストリームファイルは、ストリームバッファ14を介してMUX/DEMUX13に供給され、パケットのヘッダ情報などに基づき多重化が分離され圧縮ビデオデータと圧縮オーディオデータとが取り出される。圧縮ビデオデータは、ビデオデコーダ20に供給されデコードされて、例えばPTSに従い端子4

10

20

30

40

50

2 から出力される。圧縮オーディオデータは、オーディオデコーダ 2 1 に供給されデコードされ、ビデオデコーダ 2 0 から出力されるビデオデータと同期的に端子 4 3 から出力される。

【0270】

端子 4 2 から出力されたビデオデータは、ビデオ信号処理部 5 8 で所定の信号処理を施され、表示部 5 5 に供給される。表示部 5 5 は、供給されたビデオデータに基づく画像を映出する。端子 4 3 から出力されたオーディオデータは、音声信号処理部 5 7 で所定に信号処理され、増幅処理なども行われ、スピーカ部 6 0 に供給される。

【0271】

なお、このビデオカメラ装置 1 0 0 においても、クリップのチャプタ分割編集を行うことができる。チャプタ分割編集処理は、上述した実施の一形態による処理と何ら変わるところがないため、複雑さを避けるために説明を省略する。また、チャプタジャンプの際の処理についても、上述した実施の一形態による処理と変わらないため、説明を省略する。

10

【0272】

なお、上述では、図 3 4 に示す記録装置や図 3 6 に示すビデオカメラ装置 1 0 0 の記録再生部 1 0 がハードウェア的に構成されるように説明したが、これはこの例に限定されない。すなわち、記録再生部 1 0 は、ソフトウェアとして構成することも可能である。この場合、ソフトウェアは、例えば制御部 3 0 が有する図示されない R O M に予め記憶される。これに限らず、記録再生部 1 0 を、パーソナルコンピュータなどのコンピュータ装置上に構成することも可能である。この場合には、記録再生部 1 0 をコンピュータ装置に実行させるソフトウェアは、C D - R O M や D V D - R O M といった記録媒体に記録されて提供される。コンピュータ装置がネットワーク接続可能な場合、インターネットなどのネットワークを介して当該ソフトウェアを提供することも可能である。

20

【0273】

さらに、上述では、この発明が記録媒体にコンテンツデータなどを記録する記録装置または記録再生装置に適用されるように説明したが、これはこの例に限定されない。この発明は、記録媒体に対するコンテンツデータの記録手段や記録媒体からのコンテンツデータの再生手段を持たない、例えばコンテンツデータおよび当該コンテンツデータの再生制御情報（インデックスファイル、ムービーオブジェクト、プレイリストおよびクリップインフォメーションファイルなど）を編集する編集装置などにも適用可能なものである。この場合には、例えば、記録媒体からのデータの読み出しや記録媒体に対する書き込みは、編集装置に接続された記録再生装置により行うことが考えられる。

30

【図面の簡単な説明】

【0274】

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

【図 2】インデックステーブルを説明するための略線図である。

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

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

40

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

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

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

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

。

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

。

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

50

【図 1 1】ブロックblkPlayList()の一例の構造を表すシンタクスを示す略線図である。

【図 1 2】ブロックblkPlayItem()の一例の構造を表すシンタクスを示す略線図である。

【図 1 3】第 1 および第 2 のシームレス接続を説明するための略線図である。

【図 1 4】ブロックblkPlayListMark()の一例の構造を表すシンタクスを示す略線図である。

【図 1 5】クリップインフォメーションファイルの一例の構造を表すシンタクスを示す略線図である。

【図 1 6】ブロックblkCPI()の一例の構造を表すシンタクスを示す略線図である。

【図 1 7】ブロックblkEPMaP()の一例の構造を表すシンタクスを示す略線図である。

【図 1 8】ブロックblkEPMaPForOneStreamPID(EP_stream_type, Nc, Nf)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 1 9】エントリPTSEPCoarseおよびエントリPTSEPFineの一例のフォーマットについて示す略線図である。

【図 2 0】エントリSPNEPCoarseおよびエントリSPNEPFineの一例のフォーマットについて示す略線図である。

【図 2 1】プレイリスト以下の一例のファイル構造を示す略線図である。

【図 2 2】プレイリストを用いたクリップの分割編集について概略的に説明するための略線図である。

【図 2 3】プレイリストマークを追加挿入する一例の処理を示すフローチャートである。

【図 2 4】プレイリストマークを追加挿入する処理を説明するための略線図である。

20

【図 2 5】プレイリストマークを追加挿入する処理を説明するための略線図である。

【図 2 6】プレイリストマークをエントリPTSEPStartで示される位置に設ける方法と、GOP境界の位置に設ける方法とのそれぞれのチャプタ境界の例を示す略線図である。

【図 2 7】プレイリストマークをエントリPTSEPStartで示される位置に設ける方法と、GOP境界の位置に設ける方法とのそれぞれのデコード区間の例を示す略線図である。

【図 2 8】GOPがオープンGOPおよびクローズドGOPの何れかを判別可能な場合の、プレイリストマークをエントリPTSEPStartで示される位置に設ける方法と、GOP境界の位置に設ける方法とのそれぞれのデコード区間の例を示す略線図である。

【図 2 9】プレイリストマークの設定について各条件に関する評価を纏めて示す略線図である。

30

【図 3 0】チャプタ分割編集を行う際の、プレイリストマークの一例の挿入方法を示すフローチャートである。

【図 3 1】記録開始に応じて記録開始位置に自動的にプレイリストを設ける場合の一例の処理を示すフローチャートである。

【図 3 2】仮想プレーヤの動作を概略的に示すフローチャートである。

【図 3 3】仮想プレーヤの動作を概略的に示す略線図である。

【図 3 4】この発明の実施の一形態に適用可能な記録再生装置の一例の構成を概略的に示すブロック図である。

【図 3 5】プレイリストマークに基づくチャプタジャンプ処理の一例の方法を示すフローチャートである。

40

【図 3 6】この発明の実施の一形態の他の例によるビデオカメラ装置の一例の構成を示す。

【図 3 7】フレーム間圧縮を行ったデータのデコード処理について説明するための略線図である。

【符号の説明】

【0 2 7 5】

1 0 記録再生部

1 1 ビデオエンコーダ

1 2 オーディオエンコーダ

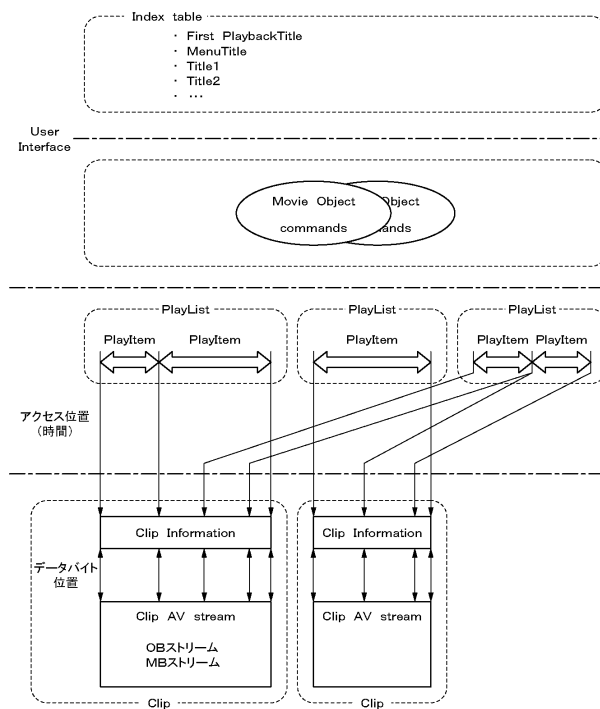
1 3 マルチプレクサ

50

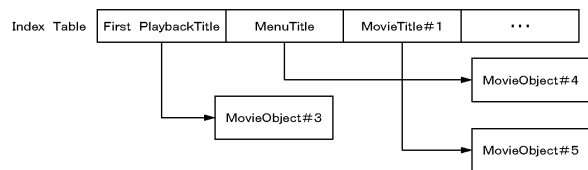
1 4 ストリームバッファ
 1 5 記録制御部
 1 6 管理情報処理部
 1 7 揮発性メモリ
 1 8 不揮発性メモリ
 2 0 ビデオデコーダ
 2 1 オーディオデコーダ
 3 0 制御部
 3 1 ユーザインターフェイス部
 3 2 記録媒体
 5 0 カメラ部
 1 0 0 ビデオカメラ装置

10

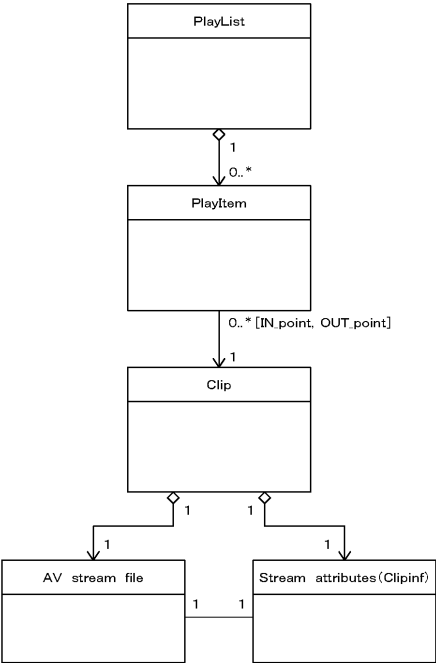
【図 1】



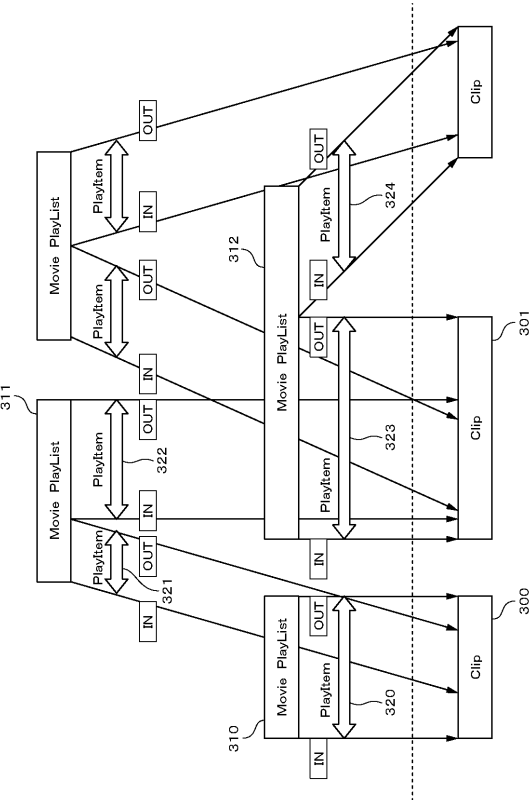
【図 2】



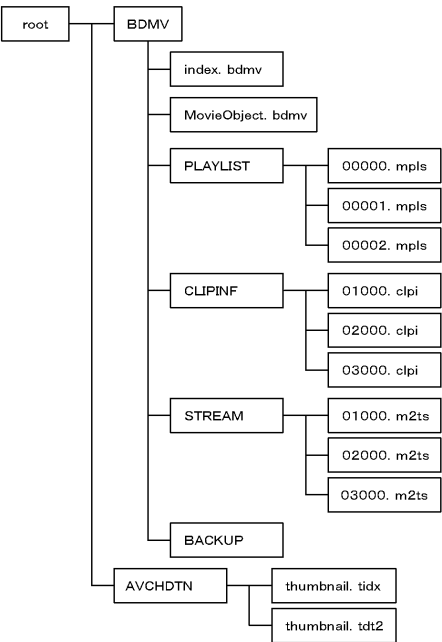
【図 3】



【図 4】



【図 5】



【図 6】

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

【図 7】

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

【図 8】

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

【図 9】

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

【図 10】

シンタクス	データ長 (ビット)	ニーモニック
Movie Playlist file {		
TypeIndicator	8 * 4	bslbf
TypeIndicator2	8 * 4	bslbf
PlaylistStartAddress	32	uimsbf
PlaylistMarkStartAddress	32	uimsbf
ExtensionDataStartAddress	32	uimsbf
reserved	160	bslbf
blkAppInfoPlaylist()		
for (i=0; i<N1; i++) {		
padding_word	16	bslbf
}		
blkPlaylist()		
for (i=0; i<N2; i++) {		
padding_word	16	bslbf
}		
blkPlaylistMark()		
for (i=0; i<N3; i++) {		
padding_word	16	bslbf
}		
blkExtensionData()		
for (i=0; i<N4; i++) {		
padding_word	16	bslbf
}		
}		

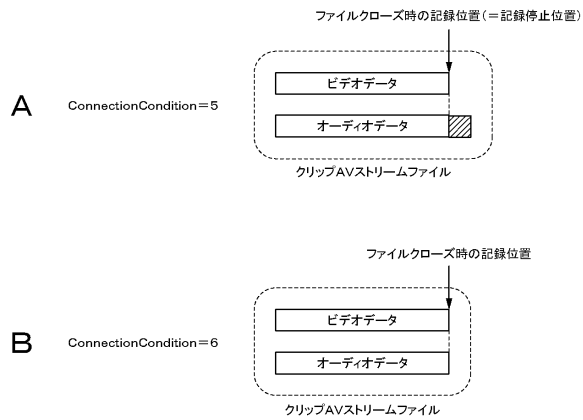
【図 11】

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

【図 12】

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

【図 13】



【図 14】

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

【図 15】

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

【図 16】

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

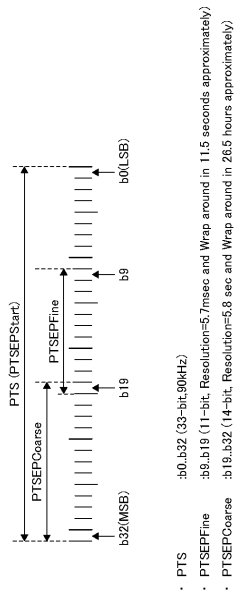
【図 18】

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

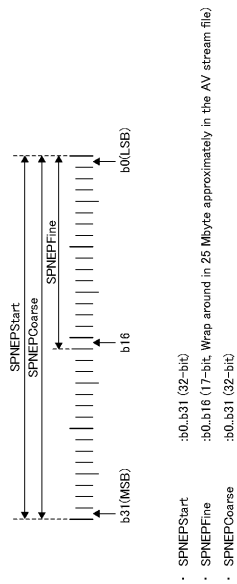
【図 17】

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

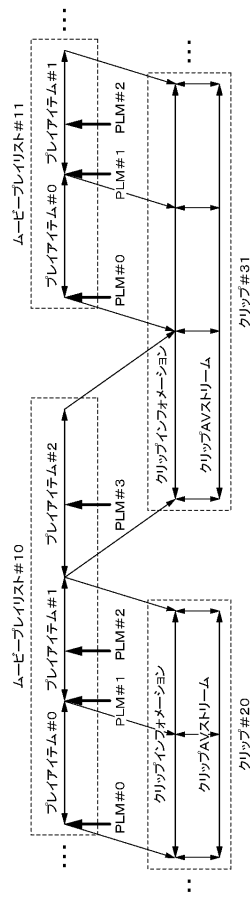
【 図 1 9 】



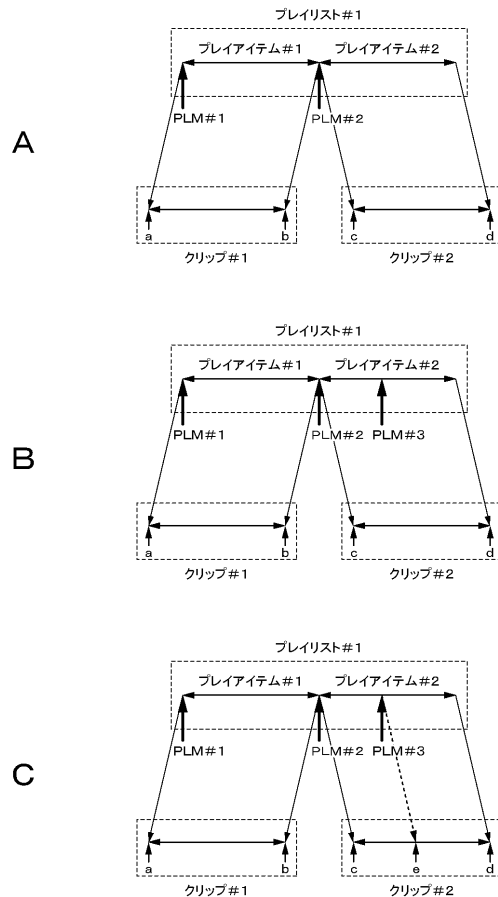
【 図 2 0 】



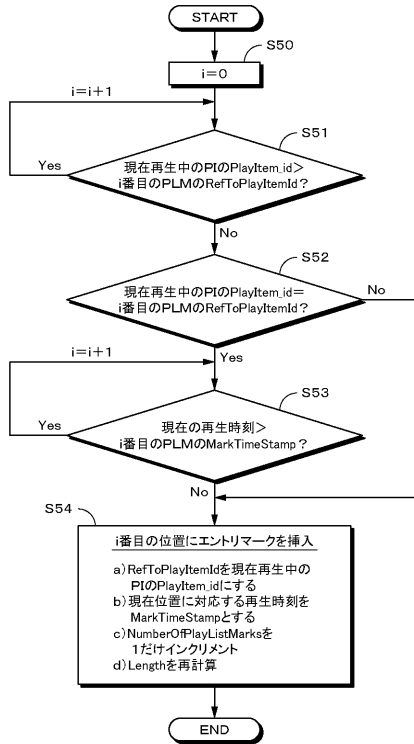
【 図 2 1 】



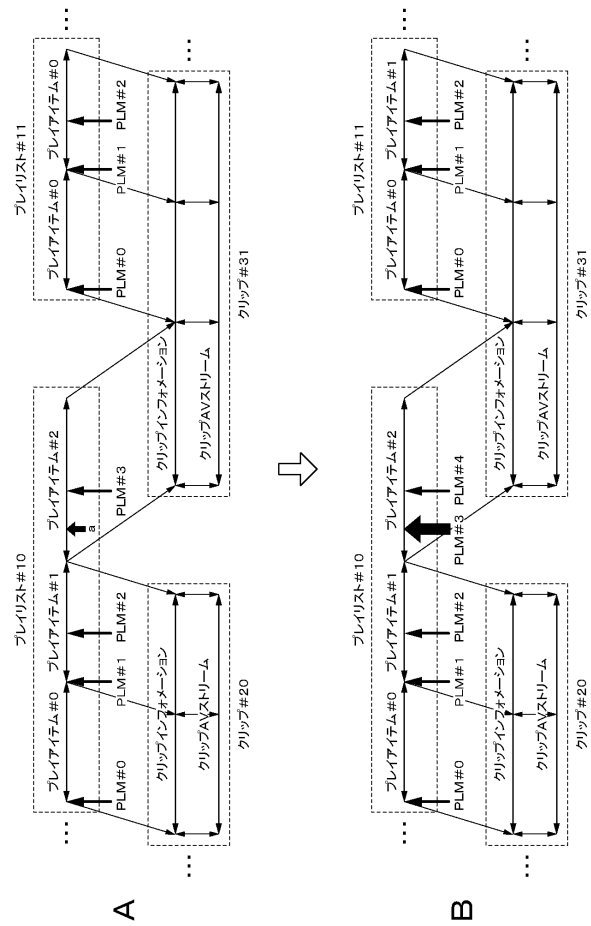
【圖 2 2】



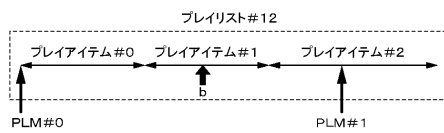
【図 23】



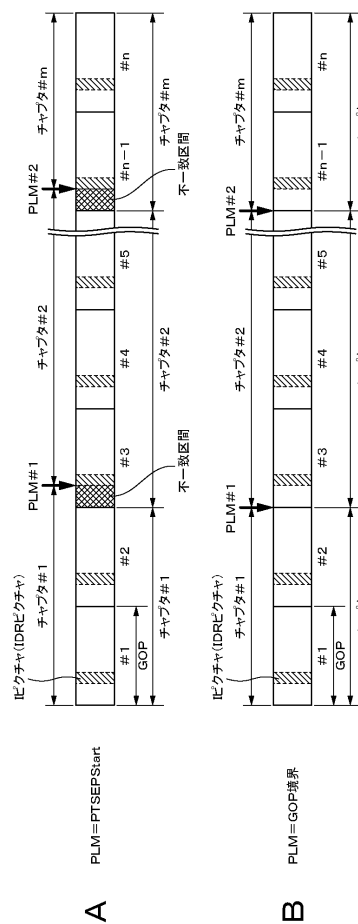
【図 24】



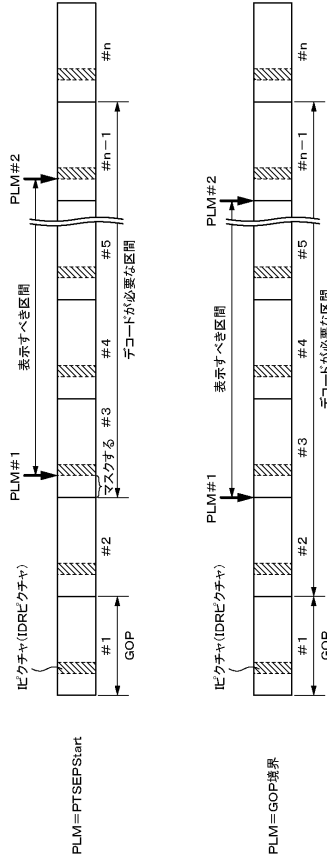
【図 25】



【図 26】



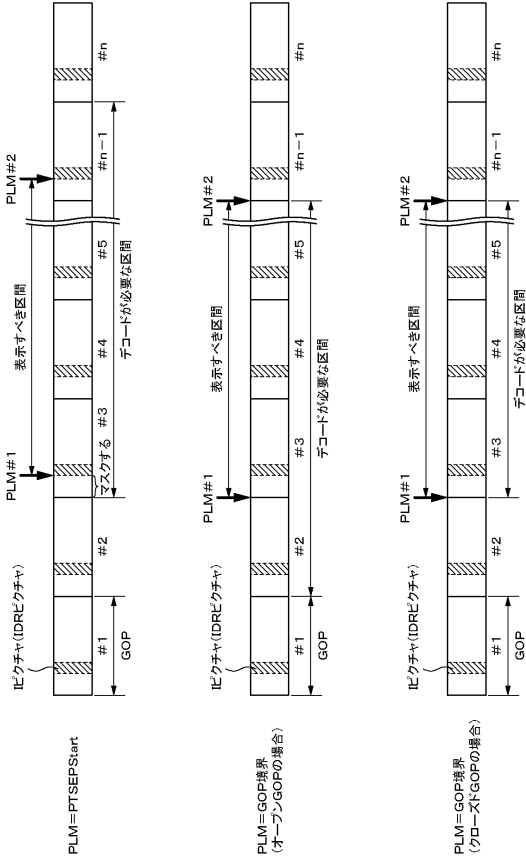
【図 27】



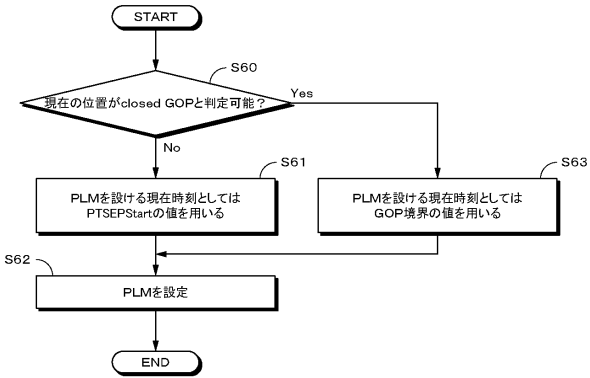
【図 29】

実源の位置 指定位置	項目	PTSEStart	GOP境界 (一般再生機)	GOP境界 (知目情報利用機)
記録start/stop 境界の場合	記録と再生とにおける チャプタ境界のすれの有無	x	○	○
	開始位置でリード・デコードが 表示フレームより多く必要な区間	○ (M-1)枚分のみ 必要	○ closedであれば 不要	○ closedであれば 不要
	終了位置でリード・デコードが 表示フレームより多く必要な区間	x (N-M+1)枚分 必要	○ 不要	○ 不要
記録start/stop 境界でない場合	記録と再生とにおける チャプタ境界のすれの有無	-	-	-
	開始位置でリード・デコードが 表示フレームより多く必要な区間	○ (M-1)枚分のみ 必要	x N枚分必要	x 非closedであれば N枚分必要
	終了位置でリード・デコードが 表示フレームより多く必要な区間	x (N-M+1)枚分 必要	x N枚分必要	○ 不要

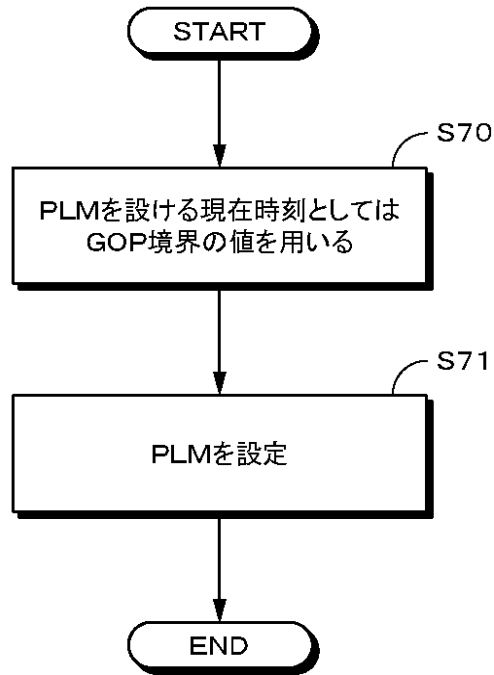
【図 28】



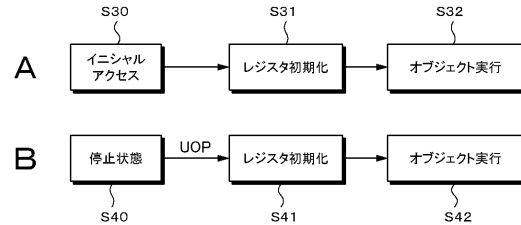
【図 30】



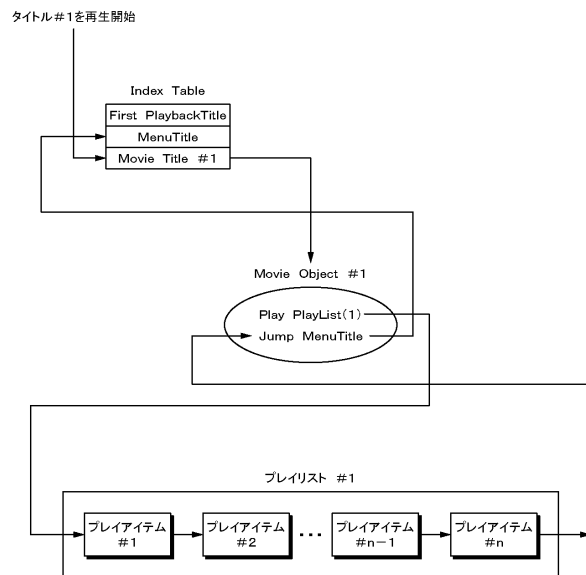
【図 3 1】



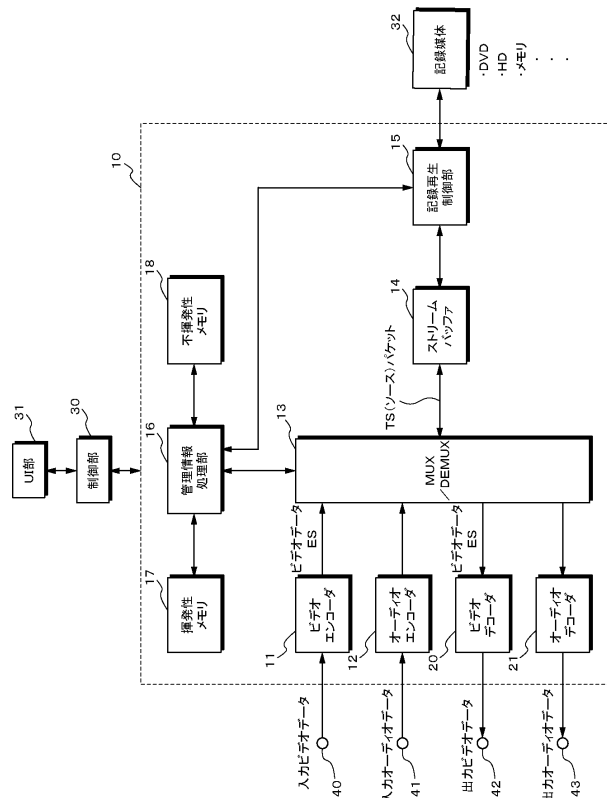
【図 3 2】



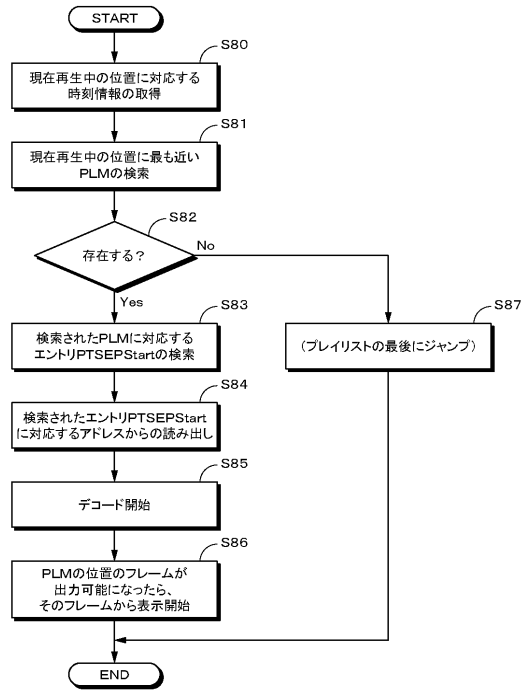
【図 3 3】



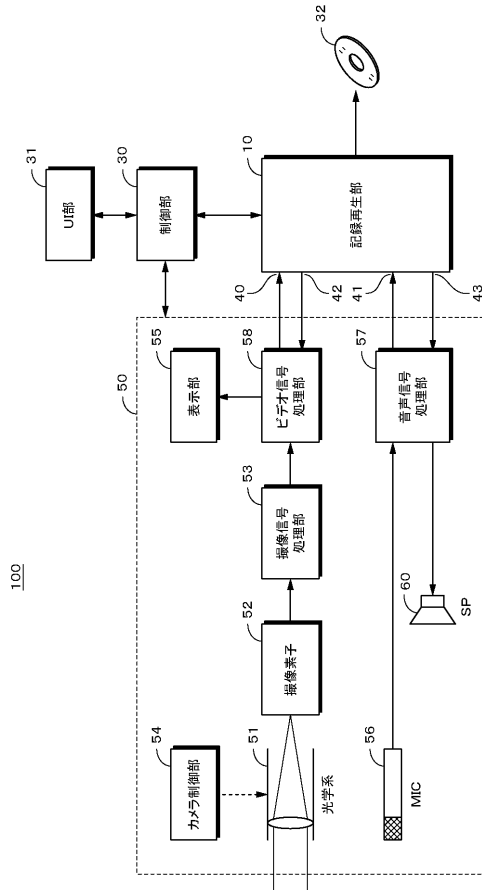
【図 3 4】



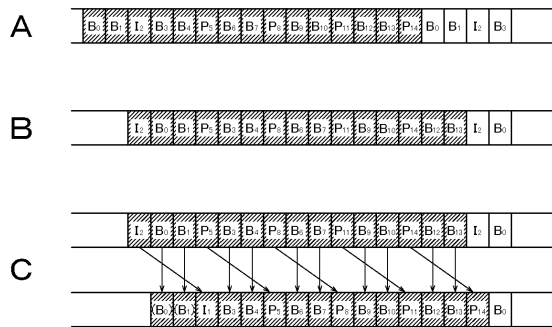
【 図 3 5 】



【 図 3 6 】



【 図 3 7 】



フロントページの続き

(51)Int.Cl.			F I		
H 0 4 N	5/91	(2006.01)	H 0 4 N	5/91	N
H 0 4 N	7/26	(2006.01)	H 0 4 N	7/13	Z

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

審査官 竹中 辰利

(56)参考文献 特表2006-506763(JP,A)
国際公開第2005/101827(WO,A1)

(58)調査した分野(Int.Cl.,DB名)

H 0 4 N	5 / 7 6 - 5 / 9 5 6
G 1 1 B	2 0 / 1 0 - 2 0 / 1 2
G 1 1 B	2 7 / 0 0 - 2 7 / 1 0
H 0 4 N	7 / 2 6