

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4481889号
(P4481889)

(45) 発行日 平成22年6月16日(2010.6.16)

(24) 登録日 平成22年3月26日(2010.3.26)

(51) Int.Cl.

F 1

G 11 B 27/02	(2006.01)
H 04 N 5/92	(2006.01)
G 11 B 20/10	(2006.01)

G 11 B 27/02	B
H 04 N 5/92	H
G 11 B 20/10	3 1 1
G 11 B 20/10	D

請求項の数 3 (全 21 頁)

(21) 出願番号	特願2005-188297 (P2005-188297)
(22) 出願日	平成17年6月28日 (2005.6.28)
(65) 公開番号	特開2007-12112 (P2007-12112A)
(43) 公開日	平成19年1月18日 (2007.1.18)
審査請求日	平成20年6月26日 (2008.6.26)

前置審査

(73) 特許権者	000001007 キヤノン株式会社 東京都大田区下丸子3丁目30番2号
(74) 代理人	100096965 弁理士 内尾 裕一
(72) 発明者	強矢 亨 東京都大田区下丸子3丁目30番2号キヤノン株式会社内

審査官 宮下 誠

最終頁に続く

(54) 【発明の名称】データ記録装置及びその方法、プログラム、記録媒体

(57) 【特許請求の範囲】

【請求項 1】

動画像データと音声データのうち少なくとも1つのメディアデータを記録するデータ記録装置であって、

前記メディアデータをフラグメント単位で記録する記録手段と、

前記フラグメントを時系列順に連結する連結手段と、

前記動画像データあるいは音声データに閾値を超える変化があった場合、前記閾値を超える変化があった動画像データあるいは音声データのメディアデータのフラグメントに、変化があったことを示すフラグをセットする設定手段と、

前記フラグメントが前記記録手段によりクラスタ境界から記録されるように制御すると共に、前記記録手段により記録されたデータ容量が所定のデータ容量に達したあとに記録されるフラグメントが、前記記録手段により記録済のフラグメントであって前記変化があつたことを示すフラグがセットされていないフラグメントのうち記録順序が最も古いフラグメントが記録されたクラスタ境界から記録されるように制御する記録制御手段と、

前記記録手段により記録されたデータ容量が前記所定のデータ容量に達したあとに前記メディアデータの先頭のフラグメントに上書きでフラグメントを記録した場合、当該フラグメントの記録後に記録されているフラグメントのうち記録順序が最も古いフラグメントのヘッダ情報を先頭のフラグメントであることを示す形式のヘッダ情報に変換するヘッダ変換手段と、

前記ヘッダ変換手段によって変換されたヘッダ情報と、前記記録手段によりクラスタ境

10

20

界から記録され前記連結手段により時系列順に連結されたフラグメントとに基づいて、ファイルを生成するファイル生成手段と
を有することを特徴とするデータ記録装置。

【請求項 2】

前記連結手段は、FAT(File Allocation Table)のクラスタ連結情報を書き換えることで、前記フラグメント単位のデータを時系列順に連結することを特徴とする請求項1に記載のデータ記録装置。

【請求項 3】

前記ファイルは、MP4におけるフラグメントムービー形であることを特徴とする請求項1又は2に記載のデータ記録装置。 10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は比較的長時間にわたるマルチメディアファイルの記録に好適なデータ記録方法および装置およびプログラムに関する。

【背景技術】

【0002】

近年、動画・音声符号化形式の多様化にともない、様々な形式を均一的な枠組みの中で相互接続可能な形で処理出来るようになる必要性が高まってきている。そこで、ISO/I E C J T C 1 / S C 2 9 / W G 1 1 (International Organization for Standardization/International Engineering Consortium)によって、M P E Gなどの動画・音声のコンテンツデータをファイルに記録するために「ISO Base Media File Format」という汎用のファイル形式が規格化されている(非特許文献1参照)。

【0003】

このファイル形式は特定の符号化形式を前提とはしていない基本ファイル形式として定義されており、所定の符号化形式や目的に適合させるにはこの規格を部分的に拡張した規格を別途定義することによって対応するといった特徴を持っている。その拡張の代表例として、M P E G - 4 の動画・音声符号化データを記録するための標準ファイル形式である「M P 4 ファイル形式」(非特許文献2参照)のようなものがある。また、同じくM P E G - 4 の動画・音声を扱う形式であるが、第三世代携帯電話を中心とする無線端末上での利用を前提に制約が課せられた動画ファイル規格として、3 G P P (Third Generation Partnership Project)によって定められた3 G P P ファイル形式(非特許文献3参照)のようなものもある。 30

【0004】

近年では、デジタルカメラや携帯電話などの機器で動画・音声データをM P E G - 4 形式で符号化してファイルに記録する際の記録形式として、上記のようなファイル形式を採用するケースが増えてきており、今後更に普及するものと考えられる。

【非特許文献1】ISO / I E C 1 4 4 9 6 - 1 2 ; “Information technology - Coding of audio-visual objects - Part 12 : ISO base media file format”; ISO / I E C ; 2 0 0 4 - 0 1 - 2 3 40

【非特許文献2】ISO / I E C 1 4 4 9 6 - 1 4 ; “Information technology - Coding of audio-visual objects - Part 14 : MP4 file format”; ISO / I E C ; 2 0 0 3 - 1 1 - 2 4

【非特許文献3】3 G P P T S 2 6 . 2 4 4 “Technical Specification Group Services and System Aspects Transparent end-to-end packet switched” 50

streaming service (PSS); 3GPP file format
(3GPP) (Release 6)" 3rd Generation Partnership Project; 2003-02-28

【発明の開示】

【発明が解決しようとする課題】

【0005】

MPEG-4は比較的高圧縮な符号化方式であり、ストリーミング配信などにおいてもエラー訂正技術などにおいて優れていることから、記録データの蓄積も考慮した監視カメラ（監視ビデオ）などへの応用も進められている。そこで、監視映像及び音声の蓄積に注目すると、長時間の監視映像及び音声を記録メディアに残したいというニーズに対しては、記録メディアに十分な容量が用意されていない場合、記録メディアの容量を使い切った時点で記録を終了されていた。

【0006】

最新のデータは残しておきたいというニーズに対しては、ファイルを分割して記録するなどして、容量が足りなくなったら古いデータを削除して新しいデータを追記するような処理方法が考えられるが、この処理方法で最新のデータを記録メディアに残す場合、ファイルがバラバラで記録されるためハンドリングを良くするためにプレイリストなどで対応付けする手法などが考えられる。しかし、この場合、他の再生処理系にデータを移動した場合にプレイリストの機能が使用できないなど、データの汎用性やハンドリングの点で問題が発生する可能性が高い。

【0007】

このように、監視ビデオのように比較的長時間の動画像や音声データを、容量に限りのある記録メディアに最新のデータを優先して記録する場合には、記録したデータの扱いの点において、汎用性やハンドリングなどの点で問題があった。

【0008】

本発明は前記の問題を解決するため、容量に限りのある記録メディアに、最新のデータを優先して残すように記録することが可能で、尚且つ記録したデータの汎用性を損なわない技術を提供することを目的とする。

【課題を解決するための手段】

【0009】

上記の目的を達成するため、本発明に係るデータ記録装置は、動画像データと音声データのうち少なくとも1つのメディアデータを記録するデータ記録装置であって、前記メディアデータをフラグメント単位で記録する記録手段と、前記フラグメントを時系列順に連結する連結手段と、前記動画像データあるいは音声データに閾値を超える変化があった場合、前記閾値を超える変化があった動画像データあるいは音声データのメディアデータのフラグメントに、変化があったことを示すフラグをセットする設定手段と、前記フラグメントが前記記録手段によりクラスタ境界から記録されるように制御すると共に、前記記録手段により記録されたデータ容量が所定のデータ容量に達したあとに記録されるフラグメントが、前記記録手段により記録済のフラグメントであって前記変化があったことを示すフラグがセットされていないフラグメントのうち記録順序が最も古いフラグメントが記録されたクラスタ境界から記録されるように制御する記録制御手段と、前記記録手段により記録されたデータ容量が前記所定のデータ容量に達したあとに前記メディアデータの先頭のフラグメントに上書きでフラグメントを記録した場合、当該フラグメントの記録後に記録されているフラグメントのうち記録順序が最も古いフラグメントのヘッダ情報を先頭のフラグメントであることを示す形式のヘッダ情報に変換するヘッダ変換手段と、前記ヘッダ変換手段によって変換されたヘッダ情報と、前記記録手段によりクラスタ境界から記録され前記連結手段により時系列順に連結されたフラグメントとに基づいて、ファイルを生成するファイル生成手段とを有する。

【0010】

また本発明に係るデータ記録方法は、動画像データと音声データのうち少なくとも1つ

10

20

30

40

50

のメディアデータをフラグメント単位で記録手段に記録するデータ記録装置が行うデータ記録方法であって、前記フラグメントを時系列順に連結する連結工程と、前記フラグメントが前記記録手段によりクラスタ境界から記録されるように制御すると共に、前記記録工程により記録されたデータ容量が所定のデータ容量に達したあとに記録されるフラグメントが、前記記録手段により記録済のフラグメントのうち記録順序が最も古いフラグメントが記録されたクラスタ境界から記録されるように制御する記録制御工程と、前記記録手段により記録されたデータ容量が前記所定のデータ容量に達したあとにフラグメントを記録した場合、当該フラグメントの記録後に記録されているフラグメントのうち記録順序が最も古いフラグメントのヘッダ情報を先頭のフラグメントであることを示すヘッダ情報に変換するヘッダ変換工程と、前記ヘッダ変換工程によって変換されたヘッダ情報と、前記記録手段によりクラスタ境界から記録され前記連結工程により時系列順に連結されたフラグメントとに基づいて、ファイルを生成するファイル生成工程とを有することを特徴とする。
。

【発明の効果】

【0011】

本発明によれば、容量に限りのある記録メディアに長時間の動画像及び音声を記録する場合などに、最新のデータを優先して、汎用性に優れたファイル形式で記録することが可能になる。

【発明を実施するための最良の形態】

【0012】

以下、本発明の実施形態を、図面を参照しながら詳細に説明する。

【0013】

なお、本発明の実施形態は処理対象のファイルがMP4ファイル形式である場合を中心に解説しているが、MP4に限らず類似のファイル形式を用いるケースに対しても適用できる。例えば、ISOではMP4と同様の基本構造を持つファイル形式規格として、「Motion JPEG 2000ファイル形式」(ISO/IEC 15444-3)や、「AVCファイル形式」(ISO/IEC 14496-15)といった標準規格が制定されている。これらの標準規格や、前述の3GPPファイル形式など、MP4で規定されるものと類似のファイル形式およびアーキテクチャが採用されている規格に対しても、本発明の一部あるいは全部を適用することが可能である。

【0014】

図1は、上記のMP4ファイル形式におけるデータ構造を説明するための概念図である。
。

【0015】

MP4ファイル101は、映像・音声データの物理的位置、時間的位置や特性情報を示すメタデータ(ヘッダ情報)102と、符号化された映像・音声データの実体を示すメディアデータ103から構成されている。

【0016】

MP4形式では、コンテンツ全体のプレゼンテーションを「ムービー」、コンテンツを構成するメディアストリームのプレゼンテーションを「トラック」と呼んでいるが、メタデータ102には、典型的には、動画像のデータ全体を論理的に取り扱うビデオトラック104と音声のデータ全体を論理的に取り扱うオーディオトラック105が含まれており、ビデオトラック104とオーディオトラック105の基本的な構成内容は、ほとんど同等のものとなっている。

【0017】

すなわち、それぞれのトラックは、実際のメディアデータの様々なメタデータ情報を記録しており、その内容がメディアデータの特性に応じて多少異なっているだけである。

【0018】

ビデオトラック104に含まれるデータは、例えば、符号化データを復号化するための所謂デコーダの構成情報や動画像の矩形サイズなどの情報が含まれ、加えて、メディアデ

10

20

30

40

50

ータが実際に記録されているファイル上の位置を示すオフセット 106 や、メディアデータのそれぞれのフレームデータ（ピクチャと呼ばれることがある）のサイズを示すサンプルサイズ 107、それぞれのフレームデータのデコード時間を示すタイムスタンプ 108 などが記録されている。

【0019】

一方、メディアデータ 103 には、符号化データの基本単位を示す「サンプル」が連続して 1 つ以上記録されている「チャンク」と呼ばれるデータ構造により、動画像のデータと音声のデータの実体が記録されている。このチャンクは、メタデータ 102 のトラックに従って、動画像のメディアデータを含むビデオチャンク 109 と音声のメディアデータを含むオーディオチャンク 310 とにより構成されている。

10

【0020】

図 1 に示す構成は、ビデオチャンク 109 とオーディオチャンク 310 が交互に記録されているように示しているが、その記録位置や順序は必ずしもこのようになっている必要はない。この例は、一般的に記録される形式の一例に過ぎない。しかしながら、このような交互の配置（インターリーブ）は、ほぼ同時刻に再生されるべき動画と音声のデータを近い位置に配置することにより、ファイルに記録されたデータのアクセス性を高めたといった効果があり、極めて一般的に見られる方法である。

【0021】

チャンクには、個々のメディアデータのサンプルがひとつ以上含まれている。例えば、図 1 に示すように、ビデオチャンク 109 には、ビデオサンプル（フレーム）311 が連続して記録される。一般的には、このビデオサンプル（フレーム）311 は、ビデオのひとつのフレームデータ（ピクチャ）に相当する。

20

【0022】

それぞれのトラックとチャンクは次のように関連付けられている。

【0023】

例えば動画像のデータの場合、ビデオトラック 104 に含まれる情報は、メディアデータ 103 に含まれるそれぞれのビデオチャンク 109 に関する情報を含んでいる。

【0024】

オフセット 106 は、ビデオチャンク 109 のそれぞれのファイル上の相対位置を示す情報のテーブルから構成されており、テーブルの個々のエントリを参照することにより、どの位置に実際のビデオチャンクが記録されていてもその位置を知ることができるようになっている。

30

【0025】

サンプルサイズ 107 は、複数のチャンク内に含まれる複数のサンプル、すなわちビデオのフレームのそれぞれのサイズをテーブルとして記載している。より正確には、個々のチャンクの中に含まれるサンプルの数を記載した情報もビデオトラック 104 の中に記載されており、これらの情報から、個々のビデオチャンク 109 の中に含まれるサンプルを正確に取得することが可能となっている。

【0026】

タイムスタンプ 108 は、個々のサンプルのデコード時間をサンプル間の差分としてテーブルに記録するようになっている。このテーブルを参照することにより、それぞれのサンプルの所謂タイムスタンプを積算時間を計算することにより取得することが可能となる。

40

【0027】

このような、トラックとチャンクの関係は、オーディオトラック 105 とオーディオチャンク 310 についても同様に成立するよう定義されている。

【0028】

これによって、MP4 ファイルおよび ISO Base Media File Formatにおいては、メタデータ 102 とメディアデータ 103 により、符号化データを必要な単位で任意の位置からタイムスタンプなどの付加情報を持つて取得することが出来

50

るようになっている。

【0029】

なお、説明を簡単にするために、ここには規格化されているすべての記録情報については記載していないことに注意しなければならない。規格化されている定義内容の詳細は、ISO/IEC 14496 の該当部分を参照することで知ることができる。

【0030】

MP4ファイル形式では、ファイルに記録されるデータは「BOX」と呼ばれるデータ構造の内部に記述され、BOXを単位としてファイルに記録される。

【0031】

図2はBOXのフィールド定義を示す図である。BOXは、次のようなフィールドから構成される。
10

【0032】

Size : size フィールド自体を含む、BOX全体のサイズ。

【0033】

Type : BOXの種類を表す4バイトのタイプ識別子。通常は4文字の英数字で表される。

【0034】

その他のフィールドはBOXによってはオプションであるため、ここでは説明を省略する。

【0035】

ファイル中に記録されるデータは、その種類によって異なるタイプのBOXに保持される。例えば、メディアデータ103は符号化データを格納するMedia Data BOX (typeフィールドの内容は‘mdat’)。以降の説明でBOXのタイプを示す識別子が用いられる場合は、そのタイプで示されるBOXを表現しているものとする)として、メタデータ102はコンテンツ全体のメタデータ情報を格納するMovie BOX (‘moov’)として記録される。
20

【0036】

また、前述のチャunkおよびサンプルに関する情報については、チャunkのファイル中のオフセット位置はChunk Offset BOX (‘stco’)、サンプルとチャunkとの対応関係はSample To Chunk BOX (‘stsc’)、各サンプルの時間情報はTime To Sample BOX (‘stts’)、各サンプルのサイズ情報はSample Size BOX (‘sts’z’)、ランダムアクセス可能なサンプルの情報はSync Sample BOX (‘stss’)、サンプルに適用される特性情報はSample Description BOX (‘stsd’)といったBOXとして、Movie BOXの内部にトラック毎に記録される。
30

【0037】

また、MP4ファイル形式では、moovにすべてのメタデータを記録する形だけではなく、メタデータを時系列順に複数の領域に分割して記録するような形式も許可している。この形式は「フラグメントムービー」(Fragmented Movie)と呼ばれている。
40

【0038】

図3に、フラグメントムービー形式のファイルの構造を示す。フラグメントムービー形式では、コンテンツのメディアデータおよびメタデータは任意の時間単位で分割することができ、分割された「フラグメント」はファイルの先頭から時系列順に記録される。例えば図3では、moov301は最初のフラグメントのメタデータを示しており、mdat302に含まれるデータに関する情報を保持する。次に出現するmoof303は2番目のフラグメントのメタデータであり、mdat304の情報を保持する、というように以下同様にして記録される。

【0039】

このように、MP4ファイル形式のファイルでは、メディアデータに関する各種属性を
50

メタデータ領域としてメディアデータと分離して保持することによって、メディアデータが物理的にどのように格納されているかに関わらず、所望のサンプルデータに容易にアクセスすることが可能になる。

【0040】

本発明における動画像及び音声データの記録方法は、図4に示すように、フラグメントムービー形式で記録メディアに書き込む際に、各フラグメントはクラスタの境界から記録を始め、ある一定時間記録したら、次のクラスタ境界から続きのフラグメントの記録を開始するという動作を繰り返す。

【0041】

即ち、`m o o v 4 0 1`はクラスタ境界から記録を始め、ある一定時間分`m o o v 4 0 1`と`m d a t 4 0 2`を記録したら、`m d a t 4 0 2`の最後から次の空き領域のクラスタ境界まで移動(407)して、次のフラグメント`m o o f 4 0 3`と`m d a t 4 0 4`の記録を行ない、このフラグメントの記録が終了したら再び次の空き領域のクラスタ境界まで移動(408)して、同様にフラグメント`m o o f 4 0 5`と`m d a t 4 0 6`の記録を行なうという動作を繰り返す。

【0042】

通常は各フラグメントの記録をクラスタ境界から始める必要は無いが、各フラグメントの記録はクラスタ境界から始ることで、高速に各フラグメントを時間順に並び替えて連結することが可能となる。この連結手段については後で詳細に説明する。

【0043】

尚、`m d a t`のデータの終端から次のクラスタ境界までの空き領域は最終的に `F r e e S p a c e B O X` ('free') に設定する必要があるが、この設定は、記録中に隨時行なっても、記録終了時にまとめて行なっても良い。

【0044】

この記録メディアには、メモリカードやハードディスク、又はDVD(Digital Versatile Disc)に代表される書き込み可能なディスクメディアなどが考えられるが、これらの記録メディアにはそれぞれ記録可能な容量に制限があるため、前記の動作で動画像及び音声データを記録していくと記録容量に達した時点で新しいデータを書き込むことが出来なくなる。

【0045】

そこで本発明では、図5のように記録したデータが、記録メディアの容量一杯や又は任意に設定した容量制限に達した時点で、先頭の`m o o v`形式のヘッダ情報を持つデータも含めて、最も古いフラグメントからデータを削除し、新しいデータはその削除した領域に記録していく。

【0046】

即ち、容量制限に達した時点で最初に削除されるデータは、通常、フラグメントムービーの先頭にあたる`m o o v`形式のヘッダ情報(メタデータ)と、その`m o o v`に対応する`m d a t`(メディアデータ)となり、このデータを削除して`m o o f`形式のメタデータとそれに対応する`m d a t`の形で新しいデータを記録する。続いて削除されるデータは`m o o f`形式で最初に記録されたメタデータとそれに対応する`m d a t`であり、削除した領域にはやはり`m o o f`と`m d a t`の形式で新しいデータを記録するという動作を繰り返すことで、常に新しいデータがファイルに上書きされていくようになる。

【0047】

なお実際には、符号化形式にも因るが、一定時間分のメディアデータのサイズは必ずしも一定のサイズにはならないため、前述のようなデータの上書き処理を行なう際には、新しく書き込むデータのサイズが、書き込み用に用意した領域のサイズより大きくなってしまわないように制御する必要がある。つまり、2つ以上のフラグメントを一度に削除して、新しいデータを書き込む領域を確保する場合もあり得る。

【0048】

記録終了時、図6のように先頭のデータ(`m o o v`形式のメタデータ)が残されている

10

20

30

40

50

場合には、`m o o v 6 0 5` 内にある全フラグメントを含む最終的なコンテンツ全体の `d u r a t i o n` をセットし、各フラグメント間にある空き領域（`6 0 1`, `6 0 2`）が `f r e e` に設定されていなければこれを設定（`6 0 3`, `6 0 4`）して、フラグメントムービー形式の MP4 ファイルを生成する。

【0049】

一方、図 7 のように先頭のデータが削除されて残っていない場合には、まず残されたフラグメントの中で時間的に最も古いデータをファイルの先頭にするため、`m o o f` 形式のメタデータを `m o o v` 形式のメタデータに変換する必要がある。以降、ファイル構成上、時間順に各フラグメントを並び替えて連結し、各フラグメントの `m o o f` に含まれるオフセット情報を正しい値にセットする必要がある。更に各フラグメント間の空き領域が `f r e e` に設定されていなければこれを設定して、フラグメントムービー形式の MP4 ファイルを生成する。
10

【0050】

（第 1 の実施の形態）

図 8 は、本実施形態における動画像及び音声のマルチメディアデータの最新データを記録メディアに残すように記録するための情報処理装置の構成を示すブロック図である。

【0051】

図 8において、`C P U 8 0 1` は `R O M 8 0 2` あるいは `R A M 8 0 3` あるいは外部記憶装置 `8 0 4` に格納された制御プログラムとして記述された記録処理を実行する。外部記憶装置 `8 0 4` によって格納された制御プログラムは、`R A M 8 0 3` にロードされ、`C P U 8 0 1` によって実行されることになる。
20

【0052】

映像入力装置 `8 1 0` から映像 `I / F 8 0 5` を介して入力された映像データ、および、音声入力装置 `8 1 1` から音声 `I / F 8 0 6` を介して入力された音声データは、メディアドライプ `I / F 8 0 8` を介してメモリカードや CD (Compact Disc)、DVD などのメディア `8 1 2` に記録される。また、ネットワーク `I / F 8 0 7` を介してネットワーク（インターネット、LAN を含む。有線、無線を問わない）越しの外部ストレージに記録しても良い。

【0053】

また、上記各構成はバス `8 0 9` により相互に通信可能に接続され、各種機能が達成される。
30

【0054】

図 9 は本発明の実施形態において、上記のマルチメディアデータの記録処理を実行する機能モジュールの構成例である。このモジュール構成は、図 8 の情報処理装置において、`C P U` が実行するマルチメディアデータ記録処理プログラムの構成要素を示す。

【0055】

図 9においてデータ記録処理部 `9 0 1` は、まず始めに動作制御に必要なパラメータとして、これから記録する各フラグメントの時間長、記録するメディアの空き容量、出力するファイルサイズの上限値、入力される動画像または音声データのビットレートなどを受け取り、記録するデータとして動画像或いは音声のデータが入力され、上記パラメータによる制御を行ないながらファイルにデータを記録していく。
40

【0056】

連結処理部 `9 0 2` は、ファイルへのデータ記録処理が終了した後、各フラグメントの順番を時系列順に整列して連結する（再構成する）機能を有する。

【0057】

ファイル出力部 `9 0 3` は、再構成されたファイルを MP4 のフラグメントムービー形式として正しいデータになるように、メタデータの変換処理や、オフセット値の補正処理などを行ない、出力ファイルとして記録メディアに出力する。

【0058】

続いて、本実施形態で行われるデータ記録処理の基本的な手順について、図 10 のフロ
50

ーチャートを用いて説明する。

【0059】

まず、ステップS101において、動作制御に必要なパラメータとして、これから記録する各フラグメントの時間長、記録するメディアの空き容量、出力するファイルサイズの上限値、入力される動画像または音声データのデータレート(ビットレート)などを受け取り、メモリ上などに保持する。

【0060】

次にステップS102において、入力されるメディアデータのデコードのために必要な情報や、ファイルの識別のために任意に入力される情報などを受け取り、また、これから生成するフラグメントムービー形式のヘッダ情報(メタデータ)を形成するために必要なメモリの初期化処理などを行なう。10

【0061】

続いてステップS103において、入力されたメディアデータをファイルに記録する処理を開始する。

【0062】

続くステップS104では、データ記録中に任意のタイミングで記録終了の指示が行なわれた場合に、ステップS105の動画記録終了処理へと処理を移行させる。動画記録終了処理の内容については図11の説明の中で後述する。

【0063】

次にステップS106において、現在記録中のフラグメントに記録されたデータの再生時間長がステップS101にて入力されたフラグメントの時間長に達したか否かの判定が行なわれる。まだ達していない場合(Yes)は、ステップS103に戻りさらにデータを記録し、達している場合(No)は、ステップS107の処理へと移行する。20

【0064】

ステップS107では、記録中のファイルサイズが出力ファイルサイズの上限値に達したか否かの判定が行なわれる。上限に達している場合(Yes)、ステップS108へ進み、記録されたファイル中の先頭を含めたフラグメントのなかで最も古いデータからデータ削除を実行し、ステップS103に戻り、データ削除を行なった領域に新しいデータの記録を実行する。この時のデータ削除は、実際にファイル中に書き込まれているデータを初期化する必要は無く、現在書き込まれている有効なフラグメントの物理的および時間軸上の位置を識別できるように、メモリ上や別ファイルなどに有効なフラグメント全ての先頭アドレスとサイズ、及び録画開始時間を記録し、隨時更新することで対応可能である。30

【0065】

一方、まだ上限に達していない場合(No)は、ステップS109において、現在のメディアデータの書き込み位置から最も近いクラスタ境界までの領域をFree Space BOX('free')に設定し、ステップS110へ移行する。

【0066】

ステップS110では、次のフラグメントの書き出し位置をクラスタ境界に設定し、ステップS103に戻り処理を継続する。

【0067】

引き続き、図10のステップS105における動画記録終了処理の内容について説明するにあたり、フラグメントムービーの先頭以外のヘッダ情報(メタデータ)を先頭のヘッダ情報(メタデータ)の形式に再構成しなければならないケースが生じる。そのため、まずフラグメントムービー形式における分割されたメタデータの、先頭とそれ以外の形式の違いについて簡単に説明する。40

【0068】

<メタデータの再構成(変換)処理>

図12は、フラグメントムービー形式のファイル構成の概略を示すものである。図において、フラグメントムービー形式のファイルの場合、最初のmoovに含まれるMovie Extends BOX('mvex')に含まれるMovie Extends

Header BOX(‘mehd’)のfragment_durationフィールドには、全フラグメントを含むコンテンツ全体のdurationをセットしなければならない。また、moofに含まれるTrack Fragment BOX(‘traf’)に含まれるTrack Fragment Header(‘tfhd’)のbase_data_offsetフィールド、およびtracfに含まれるTrack Fragment Run BOX(‘trun’)のdata_offsetフィールドには、それぞれ後続のmdatの基準オフセット、および基準オフセットからの相対オフセットをセットしなければならない。

【0069】

したがって、メタデータの再構成処理では、上記に述べたような形式の違いを考慮した変換処理を行わなければならない。

【0070】

続いて、図10のステップS105における動画記録終了処理の内容について、図11のフローチャートを用いて説明する。

【0071】

まずステップS201において、フラグメントムービー形式のMP4の先頭のヘッダ情報形式であるmoovが、有効なフラグメントとして残されているかどうかチェックする。

【0072】

moov形式のヘッダ情報(メタデータ)が残されている場合、フラグメントの削除は発生していないので、ファイル中の論理的なデータの並び順は時系列順に並んでいるので、この場合、フラグメント単位でのデータの並べ替えを行なう必要は無く、記録終了時に確定したムービー全体のdurationを設定すれば正しいMP4ファイルを生成できる。

【0073】

即ち、図6のように先頭のデータ(moov形式のメタデータ)が残されている場合には、moov605内にあるMovie Extends BOX(‘mvex’)に含まれるMovie Extends Header BOX(‘mehd’)のfragment_durationフィールドに、全フラグメントを含む最終的なコンテンツ全体のdurationをセットすることでフラグメントムービー形式のMP4ファイルを生成できる。

【0074】

一方、moov形式のヘッダ情報(メタデータ)が残されていない場合、ステップS203のヘッダ形式変換処理により、moov形式のヘッダ情報を生成する必要がある。

【0075】

即ち、図7のように先頭のデータ(moov形式のメタデータ)が削除されて残っていない場合には、記録されたmoof形式のヘッダを持つフラグメントの中で時系列的に最も古いデータをファイルの先頭にするため、そのmoof形式のヘッダをmoov形式のヘッダに変換し、以降、ステップS204において、時系列順にフラグメントを並び替えて連結する。moof形式のヘッダをmoov形式のヘッダに変換する方法、及び、時系列順にフラグメントを並び替えて連結する方法については後述する。

【0076】

続いてステップS205において、フラグメント単位の並び替えにより、各フラグメントのヘッダ情報にもともと設定されていた、ファイル中の位置を表すオフセット値が正しくない値になるため、このオフセット値を再設定する必要がある。

【0077】

即ち、各フラグメントのmoofに含まれるTrack Fragment BOX(‘traf’)に含まれるTrack Fragment Header(‘tfhd’)のbase_data_offsetフィールド、およびtracfに含まれるTrack Fragment Run BOX(‘trun’)のdata_offsetフィ

10

20

30

40

50

ールドに、それぞれ後続の m d a t の基準オフセット、および基準オフセットからの相対オフセットを正しい値に再設定する必要がある。

【 0 0 7 8 】

この時、基準オフセット (b a s e _ d a t a _ o f f s e t) には、予め各フラグメントの先頭のオフセット値を設定するようにしておくと、本実施形態のようにフラグメント単位でファイル中での配置が変更されても変更を要するのは基準オフセットのみで、相対オフセット (d a t a _ o f f s e t) の値は変更する必要が無いため、処理が容易になる。

【 0 0 7 9 】

最後に、ステップ S 2 0 6 において、ファイルをクローズし、ムービーフラグメント形式の M P 4 として正しいファイルを出力する。ステップ S 2 0 1 からステップ S 2 0 2 を経てステップ S 2 0 6 に至る処理のものは、通常のムービーフラグメント形式の M P 4 生成と同様の処理手順である。10

【 0 0 8 0 】

なお、図 1 1 のステップ S 2 0 3 とステップ S 2 0 4 は処理の順番が入れ替わっても良い。但し、ステップ S 2 0 3 のヘッダ変換処理により、基の m o o f 形式のヘッダ領域よりも変換後の m o o v 形式のヘッダ領域の方が大きくなる場合には、オフセット値に影響が出る可能性があるので、図 1 1 の通りの順番で処理した方が処理の冗長性を防ぐ意味では望ましい。

【 0 0 8 1 】

ここで、図 1 1 のステップ S 2 0 3 における m o o f から m o o v への変換処理について、図 1 3 のフローチャートを用いて説明する。20

【 0 0 8 2 】

なお説明を簡略化するため、本実施形態では処理対象ファイルのサンプルはすべて同じ再生時間を持ち（動画像データの場合、フレームレート固定。音声データの場合、サンプリングレート固定）、優先度、パディングビットなどは設定されていないものと想定する。また、同一トラックの全てのサンプルは符号化形式などにおいて同一の情報をもつものとする。

【 0 0 8 3 】

< m o o f から m o o v へのメタデータ形式変換 >

まず、ステップ S 3 0 1 において、連結処理部 9 0 2 は処理対象のフラグメントのメタデータ部に含まれる T r a c k _ F r a g m e n t _ B O X (' t f h d ') から t r a c k _ I D 、 b a s e _ d a t a _ o f f s e t 、 d e f a u l t _ s a m p l e _ d u r a t i o n を取得する。30

【 0 0 8 4 】

次に、ステップ S 3 0 2 において、新しく作成する m o o v ヘッダに含まれる T r a c k _ H e a d e r _ B O X (' t k h d ') の t r a c k _ i d に、取得した t r a c k _ I D を設定する。

【 0 0 8 5 】

次に、ステップ S 3 0 3 において、新しく作成する m o o v ヘッダに含まれる上記の t r a c k _ I D を設定したトラックに含まれる S a m p l e _ T a b l e _ B O X ¥ (' s t t s ') の s a m p l e _ d e l t a に d e f a u l t _ s a m p l e _ d u r a t i o n を設定し、 s t t s の e n t r y _ c o u n t は 1 を設定する。40

【 0 0 8 6 】

次に、ステップ S 3 0 4 において、通常複数存在するチャンク単位での処理を実行するため、ループ処理の初期化を行う。ここでは、処理対象チャンクのインデックスとして用いられる変数 i を 1 に初期化する。

【 0 0 8 7 】

次に、ステップ S 3 0 5 において、 i 番目のチャンクに対応する t r u n [i] から、 s a m p l e _ c o u n t と d a t a _ o f f s e t を取得する。この値はそれぞれ、新50

しく作成するm o o vヘッダ内の当該トラック内の*i*番目のチャンクに含まれるサンプルの数とチャンクのオフセット値に相当する。

【0088】

次に、ステップS306において、新しく作成するm o o vヘッダに含まれる上記のtrack_IDを設定したトラックに含まれるChunk_Offset_Box('stco')の*i*番目のchunk_offsetにbase_data_offset+data_offsetの値を設定する。

【0089】

次に、ステップS307において、通常1つのチャンクに複数存在するサンプル単位での処理を実行するため、ループ処理の初期化を行う。ここでは、処理対象チャンク内のサンプル番号として用いられる変数kを1に初期化する。
10

【0090】

次に、ステップS308において、*i*番目のチャンク内のk番目のサンプルのsample_sizeを取得し、新しく作成するm o o vヘッダに含まれる上記のtrack_IDを設定したトラックに含まれるSample_Size_Box('stsz')のk番目のsample_sizeに設定する。

【0091】

次に、ステップS309において、*i*番目のチャンク内のk番目のサンプルのsample_flagsを取得する。

【0092】

次に、ステップS310において、ステップS309において取得したsample_flagsがキーフレームを示しているかチェックする。チェックの結果、キーフレームであることを示している場合、ステップS311へと処理を進め、新しく作成するm o o vヘッダに含まれる上記のtrack_IDを設定したトラックに含まれるSync_Sample_Box('stss')のsample_numberに処理対象のトラックに含まれる先頭からの累計のサンプル番号を設定する。
20

【0093】

次に、ステップS312において、*i*番目のチャンクのすべてのサンプルが処理されたかチェックする。変数kが*i*番目のチャンク内のサンプル数に満たない場合は、変数kをインクリメントし、ステップS308からS312までの処理を繰り返す。
30

【0094】

次に、ステップS313において、すべてのチャンクが処理されたかチェックする。未処理のtrackが存在する場合は、変数*i*をインクリメントし、ステップS305からS313までの処理を繰り返す。

【0095】

以上の処理を変換対象の各トラックに対して行う。即ち、m o o fに含まれる全てのt f h d以下の構成情報に対して行なうことで、処理対象フラグメントのm o o f形式のメタデータをm o o v形式のメタデータに再構成することができる。

【0096】

また、m o o vを生成する際に必要な情報で、m o o fには含まれない情報が存在する。例えば、Sample_Description_Box('stsd')に含まれる“デコード固有情報”は、m o o fには含まれないので、m o o vを削除する際には、このような情報をメモリに保持しておく必要がある。即ち、連結処理部902では、すでに処理されているm o o vの内容をRAM上に保持しておき、m o o fからm o o vへの変換を行う際に、変換元のm o o fに存在しない情報をRAM上に保持されているm o o vから補充するという形の処理が行われることが求められる。
40

【0097】

<m v e xの再生成処理>

また上述の通り、フラグメントムービー形式の場合、m o o vにはm v e xを付加しなければならないが、本実施形態において、動画像及び音声データの記録中にm o o vを削
50

除したことにより、改めてm o o fをm o o vに変換することでm o o vを生成する際に、m v e xも再生成する必要がある。

【0098】

m v e xにはT r a c k E x t e n d s B O X(‘t r e x’)という、図14で示されるフィールド定義のB O Xが含まれる。各トラックを構成するすべてのサンプルが同じ属性を持つ場合に、全サンプルに適用されるデフォルト属性を保持することが出来る。そのため、各トラックの全サンプルに適用すべき属性がある場合、t r e xに値をセットするようとする。例えば本実施形態の例では、サンプルの時間間隔はすべて同一であるものとして、d e f a u l t _ s a m p l e _ d u r a t i o nフィールドにデフォルトのサンプルの時間間隔を設定するものとする。

10

【0099】

<シーケンスナンバーの整列処理>

フラグメントムービーでは、m o o fに含まれるM o v i e F r a g m e n t H e a d e r B O X(‘m f h d’)内に、通常1からスタートしてファイル中のムービー フラグメント毎に、それらの出現順に増加するs e q u e n c e _ n u m b e rが含まれる。

【0100】

本実施形態では、限られた記録容量に最新のデータを残すために、古いフラグメントのデータを新しいフラグメントのデータで上書きする処理を行なうため、s e q u e n c e _ n u m b e rの乱れが発生する。そこで、本実施形態では、最も古いm o o fのm o o vへの変換処理を実行する際、即ち、フラグメントの上書き処理により、s e q u e n c e _ n u m b e rに乱れが発生した場合には、フラグメントを時間順に並び替える際に、m o o vに変換したm o o fの次に古いm o o fデータのs e q u e n c e _ n u m b e rを1として、順に1ずつ加算したs e q u e n c e _ n u m b e rをm o o fの出現順に各々のm f h dの領域に設定し直さなければならない。

20

【0101】

続いて、図11のステップS 2 0 4の各フラグメントの時系列順の連結処理について説明する。

【0102】

<フラグメントの時系列連結処理>

30

フラグメントムービー形式のM P 4では、前述の通り、各フラグメントのシーケンスナンバーは出現順に増加しなければならない。つまり、各フラグメントのファイル先頭からの出現順序は時系列で整列されている必要がある。本実施形態では、データ記録処理部901においてファイルサイズがある閾値を超えないように、古いデータを新しいデータで上書きする処理を行なうため、ファイル中の各フラグメントのファイル先頭からの出現順序は、上書き処理が発生した時点で時系列順には並ばなくなる。そこでファイルを記録するメディアのファイルシステムとしてF A T(F i l e A l l o c a t i o n T a b l e)ファイルシステムを採用し、このF A T領域の情報を操作することにより、各フラグメントの時系列順の整列を行なう。

【0103】

40

F A Tを採用したファイルシステムでは、記憶媒体の読み書きをクラスタ(c l u s t e r)単位で扱い、1クラスタに収まらないような大きさのファイルは、複数のクラスタに分けて格納される。このときのクラスタのアドレスは連続しているとは限らないので、どの順番でクラスタが連なっているかをF A T領域に記録する。このクラスタの連なりのことをクラスタ・チェーン(C l u s t e r C h a i n)と呼ぶ。

【0104】

なおクラスタは、通常5 1 2バイトを1つの単位とするセクタ(s e c t o r)と呼ばれる基本ブロックを複数まとめたもので、フォーマットされる記録メディアの容量などにより、最適なクラスタのサイズは変化する。

【0105】

50

本実施形態では、図4に示すように、クラスタの境界から各フラグメントの記録を開始しているので、連結処理部902において、各フラグメントの書き出し位置のFATエントリを任意に書き換えることにより、ファイル中の各フラグメントの論理的な並び順を、時系列順に整列させることができる。

【0106】

この連結処理を行なうと、各フラグメントのファイル先頭からのオフセット値が変化するため、正しい値に修正する必要があるが、この処理内容については図11の説明で述べた通りである。

【0107】

(第2の実施の形態)

10

本発明の第1の実施形態では、限られた容量の記録メディアに最新の動画像及び音声データを1つのファイルとして記録する方法について述べた。

【0108】

そこで次に第2の実施の形態として、第1の実施形態の実現方法に加えて、入力された動画像データ又は音声データに、ある閾値を超える変化があった部分のデータを別ファイルとして記録メディアに書き出す手段について説明する。

【0109】

図15は、本実施形態におけるデータ記録処理部901での処理フローを示す。基本的な処理フローは、第1の実施形態における図10の場合と同様であるので、図10における処理フローと異なる点について重点的に説明を行なう。

20

【0110】

まず、ステップS401において、各フラグメントの時間長、記録するメディアの空き容量、出力するファイルサイズの上限値、入力される動画像または音声データのデータレート(ビットレート)などに加えて、入力するデータの種類に対応したイベント検出のための閾値を受け取り、RAM803や外部記憶装置804或いは記録メディア812などに保持する。このイベント検出のための閾値は、例えば動画像データの場合、ある一定時間内における動きベクトルや、明度の変化量などがあげられる。一方、音声データの場合、音の大きさ(絶対値)やある一定時間内における音の大きさの変化量などがあげられる。

【0111】

30

次にステップS402において、入力されるメディアデータのデコードのために必要な情報や、ファイルの識別のために任意に入力される情報などを受け取り、また、これから生成するフラグメントムービー形式のヘッダ情報(メタデータ)を形成するために必要なメモリの初期化処理などを行ない、続いてステップS403において、入力されたメディアデータをファイルに記録する処理を開始する。

【0112】

次にステップS404では、ステップS403でファイルに記録したメディアデータに、イベント発生を示す変化があったか否かの判定を行なう。

【0113】

このイベント検出のため、データ記録処理部902では、映像I/F805及び音声I/F806から、記録するメディアデータと共に、映像の場合は動きベクトルや一定時間内の明度の変化量、音声の場合は音の大きさや一定時間内の音の大きさの変化量などを受け取り、ステップS401で受け取った閾値との比較を行なう。

40

【0114】

閾値を超える変化があった場合、記録中のフラグメント部に変化があったことを示すフラグをセットする(ステップS409)。以下、本実施形態ではこのフラグをイベントフラグと呼ぶ。

【0115】

イベントフラグは、例えば図16のような表形式でRAM上に保持し、フラグメント毎にそのフラグメントの開始アドレスを示す基準オフセットと、そのフラグメントのfre

50

e領域まで含めた全体のサイズと共に、そのフラグメントに記録されたデータにイベントが発生したか否かを示すイベントフラグを記録する。

【0116】

ステップS405において、記録中のファイルサイズが出力ファイルサイズの上限値に達した場合(Yes)、本実施形態では、削除対象となる古いフラグメントのデータにイベントフラグがセットされているかを確認し、イベントフラグが設定されていた場合は、このフラグメントのデータは削除せず、ステップS407において、単独の別ファイルとして記録メディアに書き出す。

【0117】

この時、書き出すフラグメントがmooif形式のヘッダ情報を持つ場合は、前述した手段により、mooif形式からmooiv形式のヘッダとなるようにメタデータの変換処理を行なう。この変換処理を実行するタイミングは、最終的にファイルの記録を終了するまでの間の、いつ実行しても良い。

10

【0118】

また、フラグメント毎の設定時間が短い場合など、フラグメント1つを単独のファイルに変換しても視聴する上で短すぎる場合などは、例えばイベントフラグが設定されたフラグメントを中心として、複数のフラグメントを別ファイルとして書き出しても良い。この処理方法に関しては、本明細書では説明を割愛する。

【0119】

ステップS407において古いデータを別ファイルとして書き出した場合、更にステップS408において、その次に古いフラグメントのデータを削除し、新しいデータを記録するための領域を確保する。

20

【0120】

図15の説明において、特に説明を行なっていない部分は、第1の実施形態における図10の場合と同様である。また、説明を行った部分以外の処理や構成は、第1の実施形態と同様である。

【0121】

(他の実施の形態)

本発明は、例えば、システム、装置、方法、プログラムもしくは記憶媒体等としての実施態様をとることが可能であり、具体的には、複数の機器から構成されるシステムに適用しても良いし、また、一つの機器からなる装置に適用しても良い。

30

【0122】

本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体(または記録媒体)を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ(またはCPUやMPU)が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているオペレーティングシステム(OS)などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

40

【0123】

さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張カードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張カードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。プログラムを供給するための記録媒体としては、例えば、フロッピー(登録商標)ディスク、ハードディスク、光デ

50

イスク、光磁気ディスク、MO、CD-ROM、CD-R、CD-RW、磁気テープ、不揮発性のメモリカード、ROM、DVD(DVD-ROM, DVD-R, DVD+R, DVD-RW, DVD+RW, DVD-RAM)、HDDVD、Blu-ray Discなどがある。

【0124】

本発明を上記記憶媒体に適用する場合、その記憶媒体には、先に説明したフローチャートに対応するプログラムコードが格納されることになる。

【図面の簡単な説明】

【0125】

【図1】MP4或いは類似のファイル形式におけるメタデータ、メディアデータの概念を説明する図である。 10

【図2】BOXのフィールド定義を示す図である。

【図3】フラグメントムービーの概念を説明する図である。

【図4】本発明における基本的なデータ記録の方法を説明する図である。

【図5】本発明における最新のデータをファイルに記録する方法を説明する図である。

【図6】本発明における記録終了時の基本的なデータ処理の方法を説明する図である。

【図7】本発明における記録終了時の最新のデータを記録したファイルを正しいMP4ファイル形式に変換するための方法を説明する図である。 20

【図8】本発明の実施形態における動画像及び音声のマルチメディアデータの最新データを記録メディアに残すように記録するための情報処理装置の構成を示すブロック図である。

【図9】本発明の実施形態におけるマルチメディアデータの記録処理を実行する機能モジュールの構成例である。 20

【図10】本発明の実施形態におけるデータ記録処理の基本的な手順を示すフローチャートである。

【図11】本発明の実施形態における動画記録終了処理の手順を示すフローチャートである。

【図12】フラグメントムービー形式のファイル構成の概略を示すものである。

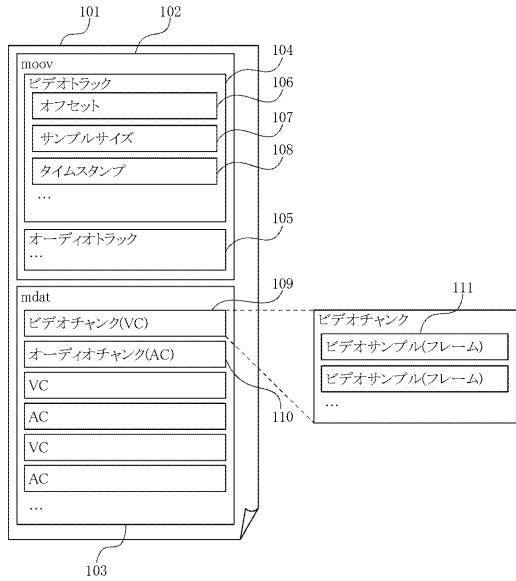
【図13】本発明の実施形態におけるmoofをmooeに変換する処理を説明するフローチャートである。 30

【図14】Track Extends Box('trex')のフィールド定義を示す図である。

【図15】本発明の実施形態におけるデータ記録処理において、イベント発生時のデータを記録する手順を示すフローチャートである。

【図16】本発明の実施形態におけるイベントフラグの記録方法を説明する図である。

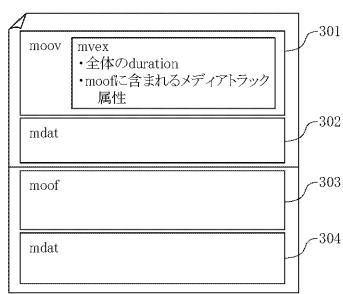
【図1】



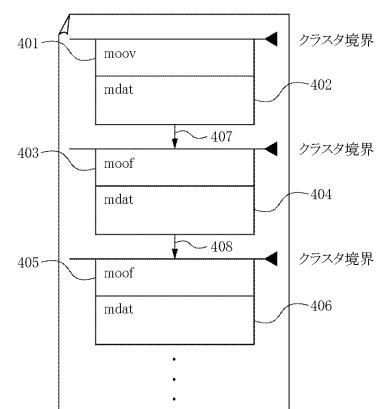
【図2】

```
aligned(8) class Box (unsigned int(32) boxtyle,
                     optional unsigned int(8)[16] extended_type) {
    unsigned int(32) size;
    unsigned int(32) type = boxtyle;
    if (size==1) {
        unsigned int(64) largesize;
    } else if (size==0) {
        // box extends to end of file
    } if (boxtyle=='uuid') {
        unsigned int(8)[16] usertype = extended_type;
    }
}
```

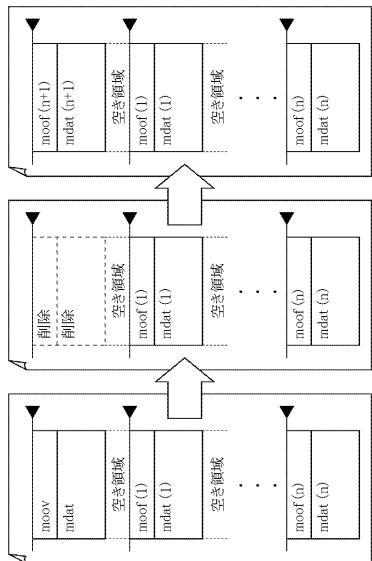
【図3】



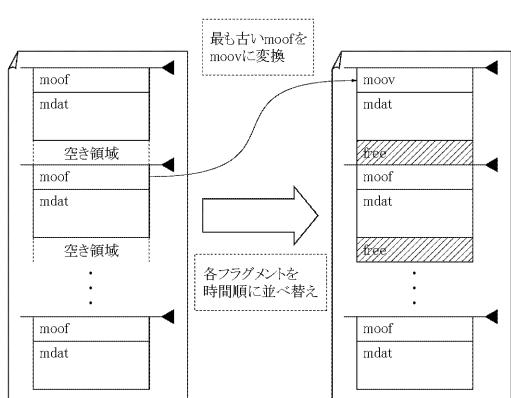
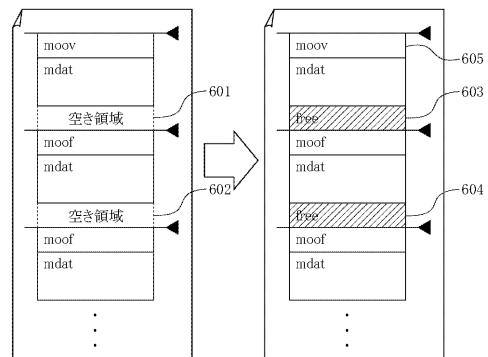
【図4】



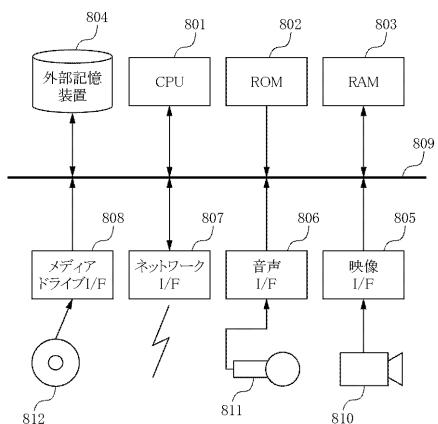
【図5】



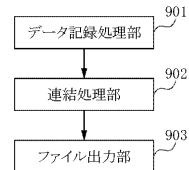
【図6】



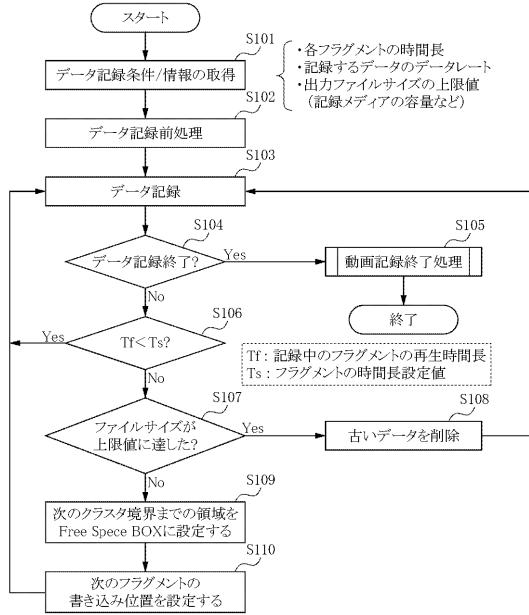
【図8】



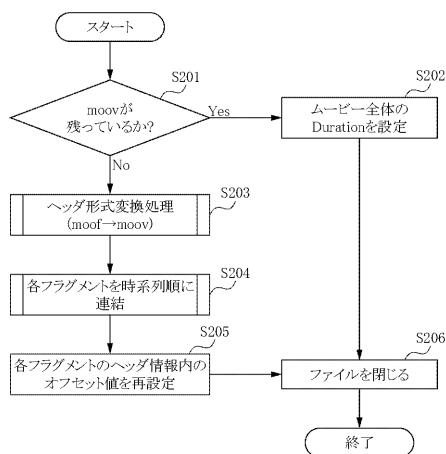
【図9】



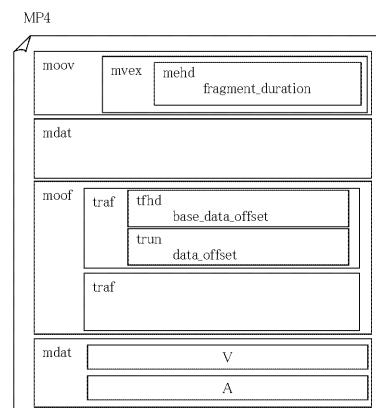
【図10】



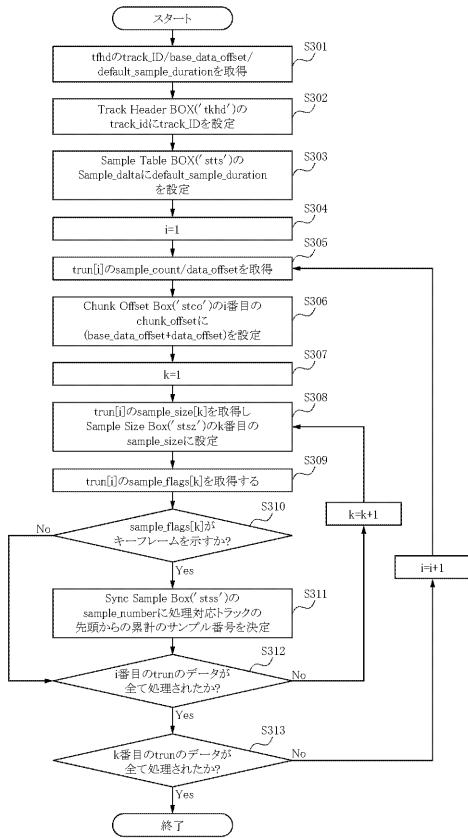
【図11】



【図12】



【図13】



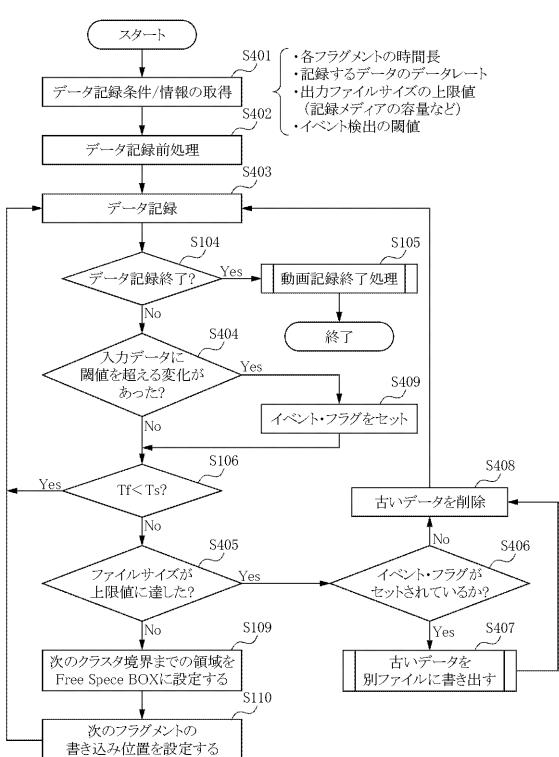
```

aligned(8) class TrackExtendsBox
    extends FullBox('trex', 0, 0) {
        unsigned int(32) track_ID;
        unsigned int(32) default_sample_description_index;
        unsigned int(32) default_sample_duration;
        unsigned int(32) default_sample_size;
        unsigned int(32) default_sample_flags;
    }
}

```

【図14】

【図15】



	開始アドレス	サイズ	フラグ
Frag.1	基準オフセット(1)	データサイズ(1)	イベントフラグ(1)
Frag.2	基準オフセット(2)	データサイズ(2)	イベントフラグ(2)
Frag.3	基準オフセット(3)	データサイズ(3)	イベントフラグ(3)
...
...

【図16】

図16

フロントページの続き

(56)参考文献 特開2000-021093(JP,A)
特開2005-079891(JP,A)
特開2000-285653(JP,A)
特開2005-117534(JP,A)
特開2004-312268(JP,A)
国際公開第01/037563(WO,A1)

(58)調査した分野(Int.Cl., DB名)

G 11 B 2 7 / 0 0
G 11 B 2 0 / 1 0
H 0 4 N 5 / 7 6
H 0 4 N 5 / 9 1