(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4280701号
(P4280701)
(45)発行日　平成21年6月17日(2009.6.17)　(24)登録日　平成21年3月19日(2009.3.19)

(51)Int.Cl.　F I
H04N 5/91 (2006.01)　H04N 5/91 N

請求項の数 9　(全 23 頁)

(21)出願番号　特願2004-314719(P2004-314719)
(22)出願日　平成16年10月28日(2004.10.28)
(65)公開番号　特開2006-129081(P2006-129081A)
(43)公開日　平成18年5月18日(2006.5.18)
審査請求日　平成19年10月24日(2007.10.24)

(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　100076428
弁理士　大塚　康徳
(74)代理人　100112508
弁理士　高柳　司郎
(74)代理人　100115071
弁理士　大塚　康弘
(74)代理人　100116894
弁理士　木村　秀二
(74)代理人　100130409
弁理士　下山　治
(74)代理人　100134175
弁理士　永川　行光

最終頁に続く

(54)【発明の名称】データファイルの編集方法及び装置及び制御プログラム及び記憶媒体

(57)【特許請求の範囲】
【請求項１】
メタデータとメディアデータとを有するデータファイルを編集するためのデータファイルの編集方法であって、
第１のデータファイルのメディアデータと前記第１のデータファイルに追加される第２のデータファイルのメディアデータが所定単位に分割されているか否か、及び、前記第１のデータファイルと前記第２のデータファイルが前記所定単位に分割されたメディアデータに対応するメタデータを有するか否かを判定する判定工程と、
前記判定工程により前記第１のデータファイルのメディアデータと第２のデータファイルのメディアデータが前記所定単位に分割されていると判定され、かつ、前記第１のデータファイルと前記第２のデータファイルが前記所定単位に分割されたメディアデータに対応するメタデータを有すると判定された場合、前記所定単位、及び、メディアデータが追加される前記第１のデータファイルの位置に応じて、前記第１のデータファイルと前記第２のデータファイルのメタデータを変更する変更工程と、
前記第１のデータファイルのメディアデータと前記第２のデータファイルのメディアデータと前記変更工程で変更されたメタデータとを有する第３のデータファイルを生成する生成工程と
を具備することを特徴とするデータファイルの編集方法。
【請求項２】
前記第１のデータファイル、及び、前記第２のデータファイル、及び、前記第３のデー

タファイルは、ＭＰ４あるいは類似のファイル形式におけるフラグメントムービー形式のデータファイルであることを特徴とする請求項１に記載のデータファイルの編集方法。

【請求項３】

前記第１のデータファイルに含まれるメディアデータの一部を削除する場合、

前記変更工程では、前記判定工程により前記第１のデータファイルのメディアデータが前記所定単位に分割されたメディアデータであると判定され、かつ、前記第１のデータファイルのメタデータが前記所定単位に分割されたメディアデータに対応するメタデータであると判定されると、削除対象のメディアデータを削除すると共に、前記所定単位、及び、前記メディアデータが削除される前記第１のデータファイルの位置に基づいて、前記第１のデータファイルのメタデータを変更することを特徴とする請求項１に記載のデータファイルの編集方法。

【請求項４】

前記変更工程では、前記判定工程により前記第１のデータファイルのメディアデータと第２のデータファイルのメディアデータが前記所定単位に分割されていると判定され、かつ、前記第１のデータファイルと前記第２のデータファイルが前記所定単位に分割されたメディアデータに対応するメタデータを有すると判定された場合、前記所定単位、及び、メディアデータが追加される前記第１のデータファイルの位置に応じて、前記第１のデータファイルのメディアデータのうち前記位置よりも後ろのメディアデータが格納される位置を示すメタデータを変更することを特徴とする請求項１に記載のデータファイルの編集方法。

【請求項５】

前記変更工程では、前記第２のデータファイルのメディアデータが前記第１のデータファイルの最初のメディアデータの前に追加される場合、メディアデータが追加される前の前記第１のデータファイルの前記最初のメディアデータに対応するメタデータの形式を、対応するメディアデータがデータファイルの最初のメディアデータであることを示す形式から、データファイルの２番目以降のメディアデータであることを示す形式に変換することを特徴とする請求項１に記載のデータファイルの編集方法。

【請求項６】

前記判定工程では、分割されたメディアデータの最初のピクチャがＩピクチャで、かつ、分割されたメディアデータに含まれるＩピクチャが１つの場合、データファイルのメディアデータが前記所定単位に分割されていると判定することを特徴とする請求項１に記載のデータファイルの編集方法。

【請求項７】

メタデータとメディアデータとを有するデータファイルを編集するためのデータファイルの編集装置であって、

第１のデータファイルのメディアデータと前記第１のデータファイルに追加される第２のデータファイルのメディアデータが所定単位に分割されているか否か、及び、前記第１のデータファイルと前記第２のデータファイルが前記所定単位に分割されたメディアデータに対応するメタデータを有するか否かを判定する判定手段と、

前記判定手段により前記第１のデータファイルのメディアデータと第２のデータファイルのメディアデータが前記所定単位に分割されていると判定され、かつ、前記第１のデータファイルと前記第２のデータファイルが前記所定単位に分割されたメディアデータに対応するメタデータを有すると判定された場合、前記所定単位、及び、メディアデータが追加される前記第１のデータファイルの位置に応じて、前記第１のデータファイルと前記第２のデータファイルのメタデータを変更する変更手段と、

前記第１のデータファイルのメディアデータと前記第２のデータファイルのメディアデータと前記変更手段で変更されたメタデータとを有する第３のデータファイルを生成する生成手段と

を具備することを特徴とするデータファイルの編集装置。

【請求項８】

請求項１乃至６のいずれか１項に記載のデータファイルの編集方法をコンピュータに実行させることを特徴とする制御プログラム。
【請求項９】
請求項８に記載の制御プログラムがコンピュータが読み取り可能に記憶されたことを特徴とする記憶媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明はマルチメディアファイルの編集処理に好適な編集方法および装置およびプログラムに関する。特にMP4あるいは類似形式のファイルのカット編集技術に関するものである。
【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する必要性が高まってきている。そこで、ISO/IEC JTC1/SC29/WG11（International Organization for Standardization/International Engineering Consortium）によって、MPEG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ファイル形式」という汎用のファイル形式が規格化されている（ISO/IEC 14496-12。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、MPEG-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「MP4ファイル形式」（ISO/IEC 14496-14。非特許文献２参照）のようなものがある。また、同じくMPEG-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、3GPP(Third Generation Partnership Project)によって定められた3GPPファイル形式（TS26.244。非特許文献３参照）のようなものもある。
【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MPEG-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、今後更に普及してくると考えられる。
【０００５】
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。
【０００６】
図１に示されるように、MP4ファイル形式のファイルは、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３０１、および映像・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３０２の大きく２つのデータ構造から構成される。
【０００７】
メディアデータ３０１は、符号化データの基本単位を示す「サンプル」が連続して複数個記録されている「チャンク」と呼ばれるデータ構造から構成されている。ファイル内には、メディアデータ３０１は、チャンクの羅列として記述される。
【０００８】
メタデータ３０２には、このチャンクのファイル中でのオフセット位置、およびチャンク内に含まれるサンプルの時間情報、サイズ情報、ランダムアクセスの可否、特性情報といった情報が記録される。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、上記の情報は各トラック毎に保持される。例えば図１のファイルは動画トラック、音声トラックの２トラックを含んでいることを示しているが、動画トラックの

チャンク３０３、３０４、およびその中に含まれるサンプルの情報は、動画トラックのメタデータ領域305に記録される。

【０００９】

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。

【００１０】

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。

【００１１】

size: sizeフィールド自体を含む、Box全体のサイズ。

【００１２】

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。

【００１３】

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。

【００１４】

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３０１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３０２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。

【００１５】

また、前述のチャンクおよびサンプルに関する情報については、チャンクのファイル中でのオフセット位置はChunk Offset Box（‘stco’）、サンプルとチャンクとの対応関係は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（‘stsz’）、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。

【００１６】

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。

【００１７】

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、moov４０１は最初のフラグメントのメタデータを示しており、mdat４０２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moof４０３は２番目のフラグメントのメタデータであり、mdat４０４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。

【００１８】

このフラグメントムービー形式は3GPPファイル形式ではサポートされていないが、後進の規格として標準化された3GPP2ファイル形式(3GPP2 C.S0050-0。非特許文献４参照)では利用を許可していることから、3GPP2準拠の無線端末を中心に広く普及してくることが考えられる。

【００１９】

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っている。
【非特許文献１】ISO/IEC 14496-12; “Information technology -- Coding of audio-visual objects -- Part 12: ISO base media file format”; ISO/IEC; 2004-01-23
【非特許文献２】ISO/IEC 14496-14; “Information technology -- Coding of audio-visual objects -- Part 14: MP4 file format”; ISO/IEC; 2003-11-24
【非特許文献３】3GPP TS 26.244 “Technical Specification Group Services and System Aspects Transparent end-to-end packet switched streaming service (PSS); 3GPP file format (3GP) (Release 6)” 3rd Generation Partnership Project; 2003-02-28
【非特許文献４】3GPP2 C.S0050-0 “3GPP2 File Formats for Multimedia Services” Version 1.0 3rd Generation Partnership Project 2; 2003-12-12
【発明の開示】
【発明が解決しようとする課題】
【００２０】
MP4ファイルに対して、所望の範囲の映像・音声データを削除したり追加したりといったいわゆる「カット編集」を行う場合、従来の方法では図４に示すように、編集対象のファイルからメタデータとメディアデータ部分を取り出し、メディアデータと対応するメタデータの削除あるいは追加といった編集処理を行ったのち、再度ひとつのファイルに多重化するという作業を行う必要があった。
【００２１】
しかしながら、従来の方法には以下にあげる課題がある。
【００２２】
第一に、従来の方法ではメタデータ部分の変更量が大きくなるという課題がある。
【００２３】
メディアデータの部分削除や追加を行うと、それにともなってメディアデータを参照するメタデータに対しても同様に追加・削除を行う必要がある。特に、メタデータにはメディアデータのオフセット情報が記録されているため、図４のようにメタデータが前置されている場合、メタデータ部分のサイズが変わると後続のメディアデータ全体のオフセット位置が変わり、結果としてメタデータとして記録されているすべてのオフセット情報を変更しなければならない。すなわち、たとえ編集が局所的であったとしてもメタデータ全体に常に大きな変更が必要となってしまい、編集時のデータ処理負荷が増大してしまっていた。
【００２４】
このように、従来の方法ではファイルのカット編集処理は非常に処理コストが高いものであり、処理速度やデータ転送速度に制限のある実行環境では実現が困難であった。
【００２５】
従って、本発明は上述した課題に鑑みてなされたものであり、その目的は、MP4あるいは類似形式のファイルのカット編集を行なう場合に、編集処理を効率的に行なえるようにすることである。
【課題を解決するための手段】
【００２６】
上述した課題を解決し、目的を達成するために、本発明に係わるデータファイルの編集方法は、メタデータとメディアデータとを有するデータファイルを編集するためのデータファイルの編集方法であって、第１のデータファイルのメディアデータと前記第１のデータファイルに追加される第２のデータファイルのメディアデータが所定単位に分割されているか否か、及び、前記第１のデータファイルと前記第２のデータファイルが前記所定単位に分割されたメディアデータに対応するメタデータを有するか否かを判定する判定工程と、前記判定工程により前記第１のデータファイルのメディアデータと第２のデータファイルのメディアデータが前記所定単位に分割されていると判定され、かつ、前記第１のデータファイルと前記第２のデータファイルが前記所定単位に分割されたメディアデータに対応するメタデータを有すると判定された場合、前記所定単位、及び、メディアデータが

追加される前記第１のデータファイルの位置に応じて、前記第１のデータファイルと前記第２のデータファイルのメタデータを変更する変更工程と、前記第１のデータファイルのメディアデータと前記第２のデータファイルのメディアデータと前記変更工程で変更されたメタデータとを有する第３のデータファイルを生成する生成工程とを具備することを特徴とする。

【００２７】

また、この発明に係わるデータファイルの編集方法において、前記第１のデータファイル、及び、前記第２のデータファイル、及び、前記第３のデータファイルは、ＭＰ４あるいは類似のファイル形式におけるフラグメントムービー形式のデータファイルであることを特徴とする。

【００２８】

また、この発明に係わるデータファイルの編集方法において、前記第１のデータファイルに含まれるメディアデータの一部を削除する場合、前記変更工程では、前記判定工程により前記第１のデータファイルのメディアデータが前記所定単位に分割されたメディアデータであると判定され、かつ、前記第１のデータファイルのメタデータが前記所定単位に分割されたメディアデータに対応するメタデータであると判定されると、削除対象のメディアデータを削除すると共に、前記所定単位、及び、前記メディアデータが削除される前記第１のデータファイルの位置に基づいて、前記第１のデータファイルのメタデータを変更することを特徴とする。

【００３２】

また、この発明に係わるデータファイルの編集方法において、前記変更工程では、前記判定工程により前記第１のデータファイルのメディアデータと第２のデータファイルのメディアデータが前記所定単位に分割されていると判定され、かつ、前記第１のデータファイルと前記第２のデータファイルが前記所定単位に分割されたメディアデータに対応するメタデータを有すると判定された場合、前記所定単位、及び、メディアデータが追加される前記第１のデータファイルの位置に応じて、前記第１のデータファイルのメディアデータのうち前記位置よりも後ろのメディアデータが格納される位置を示すメタデータを変更することを特徴とする。

【００３３】

また、この発明に係わるデータファイルの編集方法において、前記変更工程では、前記第２のデータファイルのメディアデータが前記第１のデータファイルの最初のメディアデータの前に追加される場合、メディアデータが追加される前の前記第１のデータファイルの前記最初のメディアデータに対応するメタデータの形式を、対応するメディアデータがデータファイルの最初のメディアデータであることを示す形式から、データファイルの２番目以降のメディアデータであることを示す形式に変換することを特徴とする。

また、この発明に係わるデータファイルの編集方法において、前記判定工程では、分割されたメディアデータの最初のピクチャがＩピクチャで、かつ、分割されたメディアデータに含まれるＩピクチャが１つの場合、データファイルのメディアデータが前記所定単位に分割されていると判定することを特徴とする。

【００３９】

また、本発明に係わるデータファイルの編集装置は、メタデータとメディアデータとを有するデータファイルを編集するためのデータファイルの編集装置であって、第１のデータファイルのメディアデータと前記第１のデータファイルに追加される第２のデータファイルのメディアデータが所定単位に分割されているか否か、及び、前記第１のデータファイルと前記第２のデータファイルが前記所定単位に分割されたメディアデータに対応するメタデータを有するか否かを判定する判定手段と、前記判定手段により前記第１のデータファイルのメディアデータと第２のデータファイルのメディアデータが前記所定単位に分割されていると判定され、かつ、前記第１のデータファイルと前記第２のデータファイルが前記所定単位に分割されたメディアデータに対応するメタデータを有すると判定された場合、前記所定単位、及び、メディアデータが追加される前記第１のデータファイルの位

置に応じて、前記第１のデータファイルと前記第２のデータファイルのメタデータを変更する変更手段と、前記第１のデータファイルのメディアデータと前記第２のデータファイルのメディアデータと前記変更手段で変更されたメタデータとを有する第３のデータファイルを生成する生成手段とを具備することを特徴とする。

【００５２】

また、本発明に係わる制御プログラムは、上記のデータファイルの編集方法をコンピュータに実行させることを特徴とする。

【００５３】

また、本発明に係わる記憶媒体は、上記の制御プログラムがコンピュータが読み取り可能に記憶されたことを特徴とする。

【発明の効果】

【００５４】

MP4あるいは類似形式のファイルのカット編集を行なう場合に、編集処理を効率的に行なうことが可能となる。

【発明を実施するための最良の形態】

【００５５】

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。

【００５６】

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されているが、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、上述の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。

【００５７】

また、本明細書で述べている「コンテンツデータがMP4ファイル形式で記述されている」という状態は、取り扱うデータの実体が物理的なファイルであることを示すものではない。メモリ上に記憶されているMP4ファイル形式で記述されたデータなどに対しても本発明を適用することができる。

【００５８】

本発明の実施形態では、図５で示されるように、編集対象のデータを示す編集ファイル２０４の内容の削除、あるいは編集ファイル２０４に対して追加されるデータを示す追加ファイル２０５の内容の追加・挿入の編集結果として出力される出力ファイル２０６のファイル形式をフラグメントムービー形式とし、かつ、各フラグメントを単位として追加・削除といった編集を行う。なお、編集元ファイル２０４、追加ファイル２０５、出力ファイル２０６は物理的に同一のファイルであってもよい。そのようにすることで、ファイルを構成する他のフラグメントのメタデータ部分の大幅な構成変更やサイズの変更は発生しなくなり、メタデータの編集処理はオフセット情報などの部分的な変更のみで済むため、軽量なカット編集が実現できる。

【００５９】

ただし、本発明の実施形態では、メディアデータの編集単位は、当該符号化形式において削除・追加などを行っても符号化処理が正しく行われることが保証される単位とする。上記の編集単位は、例えばMPEG-4 Videoの場合はGOV(Group Of Video Object Plane)に相当する。GOVは、図６のように、MPEG-4 VideoにおけるIピクチャから次のIピクチャの直前のピクチャまでの一連のピクチャである。MPEG-4 Videoでは、メディアデータの追加・削除処理がGOV単位で行われる場合は、P、Bピクチャのような差分符号データが正しくデコード出来るようになる。

【００６０】

本発明の実施形態の編集単位は、上記のGOVのように、差分符号化形式において追加・

削除による影響が発生しない単位となる。すなわち、MPEG-1/2の場合は、編集単位はGOP（Group Of Picture）となる。Motion JPEGの場合は差分符号化データは用いられないため、編集単位はそれぞれのピクチャとなる。

【００６１】

なお、本発明の実施形態において編集を行うことが出来るファイルは以下の条件で作成されている必要がある。

（条件１）Media Data Box(‘mdat’)には、編集単位1個分のメディアデータが格納される。

（条件２）Movie Box(‘moov’)および各フラグメントのMovie Fragment Box(‘moof’)には、編集単位１個分のメディアデータに関するメタデータが格納されている。

（条件３）各トラックのメディアデータは、同じ時間長（duration）でインターリーブされている。

【００６２】

上記の制約のため、本発明の実施形態の編集処理を行う対象となるファイルは、あらかじめ上記の条件を満たすような形で作成されていることが望ましい。上記の条件を満たさないファイルは、本発明の実施形態によるカット編集で得られる編集結果は正しいものであることは保証されない点には注意が必要である。

【００６３】

（第１の実施形態）

図７は、本発明の第１の実施形態におけるマルチメディアコンテンツの編集処理を実現する情報処理装置の構成を示すブロック図である。

【００６４】

図７において、CPU１０１はROM１０２あるいはRAM１０３あるいは外部記憶装置１０４に格納された制御プログラムとして記述された編集処理を実行する。外部記憶装置１０４によって格納された制御プログラムは、RAM１０３にロードされ、CPU１０１によって実行されることになる。

【００６５】

外部記憶装置１０４には編集処理の対象となるMP4ファイル形式で記述されたコンテンツデータを格納することができる。

【００６６】

なお、編集の対象となるMP4ファイル形式のコンテンツデータは、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）よりネットワークI/F１０７を介して取得することも可能であるし、CD（Compact Disc）やDVD（Digital Versatile Disc）等のメディア１１２に記録されたコンテンツデータをメディアドライブ１０８を介して取得し、編集処理に利用することも可能である。さらに、映像入力装置１１０から映像I/F１０５を介して入力された映像データ、および、音声入力装置１１１から音声I/F１０６を介して入力された音声データを多重化することにより作成されたコンテンツデータも、編集処理に利用してもよい。

【００６７】

また、上記各構成はバス１０９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。

【００６８】

図８は、本発明の第１の実施形態において上記の編集処理を実行する機能モジュールの構成例を示す図である。このモジュール構成は、図７の情報処理装置において、CPUが実行する編集処理プログラムの構成要素を示す。

【００６９】

図８において、ファイル解析部２０１は、カット編集対象のコンテンツデータの構造を解析し、以降の処理に必要な情報を抽出する機能を有する。ファイル解析部２０１には、入力データとして、編集対象のコンテンツデータを示す編集ファイル２０４、および、データの追加を行う場合は、追加されるコンテンツデータを示す追加ファイル２０５が入力

される。なお、本実施形態では、編集ファイル２０４はフラグメントムービー形式のMP4ファイル形式で記述されているものとする。また、追加ファイル２０５は、フラグメントムービー形式のMP4ファイル、またはその一部のフラグメントとする。

【００７０】

編集制御部２０２は、ファイル解析部２０１によって取得された情報に基づいてメタデータの編集およびデータの出力指示を行う機能を有する。編集制御部２０２は、それ自体はデータの出力は行わず、ファイル出力部２０３に対して出力指示を行うことによって出力処理を行う。

【００７１】

ファイル出力部２０３は、編集制御部２０２からの出力指示にしたがって、ファイルデータの出力を行う。本実施形態では、ファイル出力部２０３は、編集結果のコンテンツデータを出力ファイル２０６として出力する。

【００７２】

次に、本実施形態で行われるカット編集処理の基本的な手順について、図９のフローチャートを用いて説明する。

【００７３】

まず、ステップＳ１において、ファイル解析部２０１は入力された編集ファイル２０４の内部構造を解析する。また、データの追加処理を行う場合は、ファイル解析部２０１に与えられる追加ファイル２０５の内部構造も同様に解析する。解析処理の結果、および解析によって得られたデータの内容は編集制御部２０２に渡され、後続の処理で利用される。

【００７４】

次に、ステップＳ２で、解析されたファイルが処理可能であるかどうかチェックする。チェック処理の手順については、後述の図１０のフローチャートを用いて説明する。対象のファイルが処理可能でなければ、後続の処理はスキップし何らかの代替処理を行う。この場合の代替処理は、編集処理が行えない旨のエラーを出力したり、従来の方法で編集処理を行うといった、本発明の主題とは異なる例外処理を想定している。

【００７５】

ファイルが処理可能であれば、次に、ステップＳ３で、編集ファイル２０４の最初のフラグメントを処理対象のフラグメントとし、以降の処理でフラグメント単位での編集処理を行う。

【００７６】

次に、ステップＳ４で、処理対象のフラグメントに対して行われる編集処理の種類によって条件分岐を行う。処理対象のフラグメントの削除を行う場合はステップＳ５の削除処理を行い、処理後はステップＳ８の処理に移行する。フラグメントの削除処理の詳細については後述する。処理対象のフラグメントの直前に追加ファイル２０５のデータの追加・挿入を行う場合はステップＳ６の追加処理を行い、処理後は引き続きステップＳ７の処理を実行する。フラグメントの追加処理の詳細については後述の図１５のフローチャートを用いて後述する。

【００７７】

処理対象のフラグメントに関連して上記いずれかの編集処理が行われない場合は、ステップＳ７で示される処理対象のフラグメントの内容を出力ファイル２０６に出力する処理を行う。フラグメントの出力処理の詳細については後述する。

【００７８】

次に、ステップＳ８で、編集ファイル２０４に未処理のフラグメントがまだ残っているかどうかをチェックする。残っていなければ、本フローチャートの一連の処理を終了する。

【００７９】

未処理フラグメントが残っている場合は、ステップＳ９として、編集ファイル２０４の最初の未処理のフラグメントを処理対象のフラグメントとして、ステップＳ４からの処理

を繰り返す。

【００８０】

続いて、図９の編集処理のステップＳ２において行われる入力チェック処理の手順を、図１０のフローチャートを用いて説明する。

【００８１】

＜ファイルのチェック処理＞

ファイル解析部２０１は、入力された編集ファイル２０４および追加ファイル２０５が本実施形態で処理可能な構造になっているかどうかをチェックする。

【００８２】

まず、ステップＳ１１において、ファイル解析部２０１は編集ファイル２０４が正しい構成を備えたMP4ファイル形式であるかチェックし、編集ファイル２０４が正しい形式でなければエラー処理を行う。なお、MP4ファイルに必要な要件は、関連する規格書に記載されているため、ここでは詳細なチェック内容についての説明は省略する。

【００８３】

次に、編集ファイル２０４に対して追加・挿入処理が行われる場合は、ステップＳ１２において、追加ファイル２０５が正しい構成を備えたMP4ファイル形式であるか、あるいはフラグメントムービー形式におけるフラグメントの集合であるかチェックし、追加ファイル２０５が正しい形式でなければエラー処理を行う。追加・挿入処理が行われない場合は、このステップの処理は実行されない。なお、MP4ファイルの要件と同様、フラグメントの仕様も関連する規格書に記載されているため、ここでは詳細なチェック内容についての説明は省略する。

【００８４】

次に、ステップＳ１３において、前述の処理可能ファイルの（条件１）、（条件２）で示されるように、編集ファイル２０４および追加ファイル２０５に含まれるmdatに、そのメディアデータの符号化形式における編集単位のデータが格納されているかどうかチェックする。各mdatに編集単位のデータが格納されていなければエラー処理を行う。編集単位のデータが格納されているかどうかのチェック方法は符号化形式によって異なるが、ここではMPEG-4 Videoを例に説明する。

【００８５】

MPEG-4 Videoでは編集単位は図６に示す通りGOVとなるが、MP4ファイルでは通常、Iピクチャはシンク・ポイントであることを示すシンク・サンプル（Sync Sample）に設定される。フラグメントムービー形式の場合、GOV単位でメディアデータがmdatに格納されていれば、そのmdatの最初のサンプルは常にシンク・サンプルとなる。また、mdatに常に1個の編集単位のデータが保持されるのであれば、シンク・サンプルの数は常に１個となる。すなわち、Sync Sample Box(‘stss’)の内容から、すべてのmdatのシンク・サンプルが最初のサンプルに設定されており、かつ、各mdatにシンク・サンプルが1個のみである場合は、mdatは1個の編集単位のデータで構成されていると考えられる。

【００８６】

また、編集単位がある固定の時間長(duration)を持っていることが必要であれば、同時にそのチェックも行う。例えば、編集単位が1秒間であることが望まれる場合、各トラックのdurationが1秒間であれば条件を満たしていると考えられる。

【００８７】

次に、ステップＳ１４において、前述の処理可能ファイルの（条件３）で示されるように、編集mdat内の各トラックのメディアデータのdurationがすべて同一であるかをチェックし、同一でなければエラー処理を行う。上記の同一であるという状態を例示すると、コンテンツデータがビデオ、オーディオ２つのトラックから構成される場合、図１１に示すように両者のメディアデータは同一のdurationを持った形でmdatに格納されている状態であることを示す。同一であれば、各トラックのメディアデータは均一な時間間隔でインターリーブされた状態でmdatに格納されている状態であるため、フラグメント単位での編集処理をすべてのトラックに対して等しく行えるようになる。

【００８８】
なお、ステップＳ１３、Ｓ１４の処理は必須ではなく、省略しても本実施形態の編集処理を行うことは可能である。ただしその場合、編集後のメディアデータの状態が適切であるかどうかは保証されないため、再生が正常に行われない場合がある点には注意が必要である。
【００８９】
引き続き、図９の編集処理のステップＳ５、Ｓ６、Ｓ７で示される各種処理を説明するにあたって、場合によっては通常のムービーのメタデータ形式をフラグメントムービーのメタデータ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、通常のムービーとフラグメントムービーのメタデータの形式の違いについて簡単に説明する。
【００９０】
＜メタデータのmoof形式への再構成処理＞
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、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(‘trhd’)のbase_data_offsetフィールド、およびtrafに含まれるTrack Fragment Run Box(‘trun’)のdata_offsetフィールドに、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。
【００９１】
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。
【００９２】
上記の違いを踏まえて、編集制御部２０２において、MP4ファイルのmoov部分を、フラグメントムービー形式のファイルで用いられるmoof形式に変換する手順について、図１３のフローチャートを用いて以下に説明する。なお、本実施形態では、追加するファイルのmdat全体をそのまま１つのフラグメントランとして変換する場合を想定する。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、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。すなわち、ここでの変換処理の対象となるメタデータは、Sample Size Box(‘stsz’)、Chunk Offset Box(‘stco’)、Sync Sample Box(‘stss’)のみとする。各サンプルの再生時間の設定処理は後述する。
【００９３】
＜moovからmoofへのメタデータ形式変換＞
まず、ステップＳ３０１において、編集制御部２０２は処理対象のフラグメントのmdatのファイル先頭からのオフセット位置を取得あるいは計算する。
【００９４】
次に、ステップＳ３０２において、以降のチャンクに対する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。
【００９５】
次に、ステップＳ３０３において、ｉ番目のチャンクに対応するフラグメントランの内容を保持するエントリであるtrun[i]を作成する。このtrun[i]は、フラグメントランの内容がファイルに出力されるまでの間、RAMに保持される。
【００９６】
次に、ステップＳ３０４において、ｉ番目のチャンクのオフセット位置をstcoから取得し、trun[i]のdata_offsetフィールドに（mdatのオフセット－ｉ番目のチャンクのオフセット）の値をセットする。stcoのchunk_offsetフィールドはファイル先頭から当該チャンクまでのオフセット値をチャンク毎に格納しているが、フラグメントムービーでは、各フ

ラグメント毎にフラグメントランまでのただ１つのオフセットと各サンプルのサイズによってサンプルの位置を確定するため、chunk_offsetの値はtrunのdata_offsetに格納されればよい。なお、data_offsetフィールドが用いられる場合、trunのtr_flagsにはdata_offset-presentビット(0x000001)をセットしなければならない。

【００９７】

次に、ステップＳ３０５において、ｉ番目のチャンクのサンプル数をSample To Chunk Box(‘stsc’)から取得し、trun[i]のsample_countフィールドにセットする。

【００９８】

次に、ステップＳ３０６において、以降のサンプルに対するループ処理の初期化を行う。ここでは、処理対象サンプルのインデックスとして用いられる変数ｊを1に初期化する。

【００９９】

次に、ステップＳ３０７において、ｉ番目のチャンク内のj番目のサンプルのサイズをstszから取得し、trun[i]のsample_size[j]フィールドをセットする。なお、sample_sizeフィールドが用いられる場合、trunのtr_flagsにはsample-size-presentビット(0x000200)をセットしなければならない。

【０１００】

次に、ステップＳ３０８において、ｉ番目のチャンク内のj番目のサンプルのランダムアクセス可否をstssから取得し、trun[i]のsample_flags[j]フィールドをセットする。なお、sample_flagsフィールドが用いられる場合、trunのtr_flagsにはsample-flags-presentビット(0x000400)をセットしなければならない。

【０１０１】

次に、ステップＳ３０９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j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ｊをインクリメントし、ステップＳ３０７からＳ３０９までの処理を繰り返す。

【０１０２】

次に、ステップＳ３１０において、すべてのチャンクのサンプルが処理されたかチェックする。変数iがstcoで示されるチャンク数に満たない場合は、変数ｉをインクリメントし、ステップＳ３０３からＳ３１０までの処理を繰り返す。

【０１０３】

以上の処理を変換対象の各トラックに対して行うことで、処理対象フラグメントのmoovの内容がmoof形式に再構成された形でRAM内に保持される。

【０１０４】

＜mvexのセット処理＞

moovからmoofへのメタデータ形式変換を行った後は、上述のmvexに必要なデータをセットする。本実施形態では、各編集処理によってコンテンツ全体のdurationが変動した場合、そのdurationをmoovに含まれるmvexにセットするようにする。

【０１０５】

さらに、mvexにはTrack Extends Box（’trex’）という、図１４で示されるフィールド定義のBox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、trexに値をセットするようにする。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、default_sample_duration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。

【０１０６】

上記に挙げたようなメタデータの再構成処理、およびそれにともなって必要となるmvexのセット処理を行うことで、通常形式のmoovからフラグメントムービー形式のmoofへの変換を規格に矛盾することなく行うことができる。

【０１０７】

同様の処理を逆に行うことによって、moofを通常形式のmoovに変換することも可能であ

る。ただし、moofからmoovへの再構成を行う際には、moovにはSample Description Box('stsd')のように、moofには存在しない情報が含まれていることについて考慮する必要がある。そのため、編集制御部２０２では、すでに処理されているmoovの内容をRAM上に保持しておき、moofからmoovへの変換を行う際に、変換元のmoofに存在しない情報をRAM上に保持されているmoovから補充するという処理が行われることが求められる。なお、本明細書では逆変換の詳細な説明は割愛する。

【０１０８】

引き続き、図９の編集処理のステップＳ５の削除処理について説明する。

【０１０９】

＜フラグメントの削除処理＞

処理対象のフラグメントの削除は、編集制御部２０２で、そのフラグメントを構成するmoovあるいはmoofとmdatを出力ファイル２０６にコピーせずに読み飛ばす。そのようにすることで、出力ファイル２０６は削除されるフラグメントは含まれない状態で生成される。

【０１１０】

＜フラグメントの追加処理＞

引き続き、図９の編集処理のステップＳ６の追加処理について、図１５のフローチャートを用いて説明する。

【０１１１】

まず、ステップＳ４０１において、編集制御部２０２で、追加を行う対象のフラグメントを追加ファイル２０５の最初のフラグメントに設定する。

【０１１２】

次に、ステップＳ４０２において、上記の追加対象のフラグメントがmoovであるかチェックする。

【０１１３】

追加対象フラグメントがmoovなら、ステップＳ４０３において、図９の処理における編集ファイル２０４の処理対象フラグメントがmoovであるかチェックする。

【０１１４】

追加対象フラグメントがmoovで、処理対象フラグメントがmoovでなければ、追加対象のフラグメントは最終的にファイルの先頭に配置されmoovとして出力されるため、何も変更せずにステップＳ４０８の処理に移行する。

【０１１５】

追加対象フラグメントがmoovで、処理対象フラグメントがmoovなら、追加対象のmoofは最終的にmoofとして出力されるため、ステップＳ４０５において、上述のmoofへの再構成処理を行う。

【０１１６】

追加対象フラグメントがmoovでなければ、ステップＳ４０４において、図９の処理における編集ファイル２０４の処理対象フラグメントがmoovであるかチェックする。

【０１１７】

追加対象フラグメントがmoovでなく、処理対象フラグメントもmoovでなければ、追加対象のフラグメントはそのままmoofとして出力される。しかし、配置位置が変わるため、ステップＳ４０６において追加対象フラグメントのmoofのオフセットを変更する処理を行う。

【０１１８】

ここでは出力ファイル２０６の出力位置に合わせて、追加対象フラグメントのtfhdのbase_data_offsetおよびtrunのdata_offsetに適切な値をセットしておく。処理後はステップＳ４０８の処理を実行する。

【０１１９】

追加対象フラグメントがmoovでなく、処理対象フラグメントがmoovなら、追加対象のmoofは最終的にファイルの先頭に配置されmoovとして出力されるため、ステップ４０７にお

いて、上述のmoovへの再構成処理を行う。処理後はステップＳ４０８の処理を実行する。
【０１２０】
次に、ステップＳ４０８において、追加対象フラグメントのメタデータおよびメディアデータを、出力ファイル２０６の末尾の位置に出力する。
【０１２１】
次に、ステップＳ４０９において、追加ファイル２０５に未処理のフラグメントがあるかチェックする。未処理の追加対象フラグメントがなければ本フローの一連の追加処理を終了する。
【０１２２】
未処理の追加対象フラグメントがある場合は、ステップＳ４１０において、次に処理される追加対象フラグメントを最初の未処理のフラグメントに設定し、ステップＳ４０２からの処理を繰り返す。
【０１２３】
＜フラグメントの出力処理＞
引き続き、図９の編集処理のステップＳ７の出力処理について、図１６のフローチャートを用いて説明する。
【０１２４】
まず、ステップＳ５０１において、編集制御部２０２で図９の処理における処理対象フラグメントがmoovであるかチェックする。
【０１２５】
処理対象フラグメントがmoovなら、ステップＳ５０２において、出力ファイル２０６に他のフラグメントがすでに出力されているかチェックする。
【０１２６】
処理対象フラグメントがmoovで、他のフラグメントが出力済みでなければ、追加対象のフラグメントは最終的にファイルの先頭に配置されmoovとして出力されるため、何も変更せずにステップＳ５０７の処理に移行する。
【０１２７】
処理対象フラグメントがmoovで、他のフラグメントが出力済みなら、追加対象のmoofは最終的にmoofとして出力されるため、ステップＳ５０４において、上述のmoofへの再構成処理を行う。
【０１２８】
処理対象フラグメントがmoovでなければ、ステップＳ５０３において、出力ファイル２０６に他のフラグメントがすでに出力されているかチェックする。
【０１２９】
処理対象フラグメントがmoovでなく、他のフラグメントが出力済みなら、追加対象のフラグメントはそのままmoofとして出力される。しかし、配置位置が変わるため、ステップＳ５０５において追加対象フラグメントのmoofのオフセットを変更する処理を行う。
【０１３０】
ここでは出力ファイル２０６の出力位置に合わせて、追加対象フラグメントのtfhdのbase_data_offsetおよびtrunのdata_offsetに適切な値をセットする。処理後はステップＳ５０７の処理を実行する。
【０１３１】
処理対象フラグメントがmoovでなく、他のフラグメントが出力済みでないなら、追加対象のmoofは最終的にファイルの先頭に配置されmoovとして出力されるため、ステップ５０６において、上述のmoovへの再構成処理を行う。処理後はステップＳ５０７の処理を実行する。
【０１３２】
次に、ステップＳ５０７において、追加対象フラグメントのメタデータおよびメディアデータを、出力ファイル２０６の末尾の位置に出力する。
【０１３３】

以上述べたようなフラグメントの削除、追加、出力処理を、図９の処理手順において処理対象のフラグメントすべてに対して行うことによって、最終的に編集結果として出力ファイル２０６が得られる。この出力ファイル２０６を得るために必要なデータ処理のほとんどは単純なデータコピーとメタデータのオフセット変更である。場合によってはフラグメントムービー形式と通常形式のメタデータ変換が発生するが、変換が必要なのは最初のフラグメントのみであり、かつ、編集単位が短時間であればデータの変換に要する時間や負荷も小さくて済む。

【０１３４】

このようにして、本実施形態では軽量で簡易なMP4ファイルのカット編集処理が実現される。

【０１３５】

本実施形態の以降の説明では、上記の手順を用いて行う編集処理の具体例を示す。なお、以降の例では、各フラグメントのdurationは1秒間とする。すなわち、編集対象のファイルに対しては「秒単位の編集」が可能である。

＜編集の具体例１＞

第１の例として、本実施形態によってファイルの一部をカットする簡易なカット編集の例について説明する。なお、簡単のため、

・タイムスケールは1
・編集ファイル２０４の全体のdurationは3秒
・各フラグメントのサイズは100kB
・各フラグメントのtfhdのbase_data_offsetは0

とする。

【０１３６】

上記のファイルに対して、図１７で示すように、編集ファイル２０４の先頭から１～２秒（６０２)の部分をカットする場合を説明する。

上記の例では、各編集単位のデータコピー以外のデータの変更内容は以下の点のみとなる。

（１）mehdのfragment_durationは、3から2に変更される
（２）フラグメント６０３のtrunのdata_offsetは、は元の値-100に変更される。

【０１３７】

以上の編集内容が反映された形で出力ファイル２０６が生成される。

【０１３８】

この様に、本実施形態によれば極めて簡易なデータ計算とデータの変更、そして編集内容に基づくデータコピーのみで、MP4ファイルのカット編集が可能になる。

【０１３９】

このケースでは各フラグメントのサイズを同一(100kB)としたが、各サイズが別々であっても同様に簡易な編集は可能である。

【０１４０】

＜編集の具体例２＞

第２の例として、本実施形態によって、編集対象のファイルに別のファイルの一部のデータを挿入する処理の例について説明する。データの追加についても、同様の仕様で作成されたMP4ファイル同士であれば、上記カット編集と同様の簡易な方法で追加・挿入処理を行なうことが出来る。なお、簡単のため

・タイムスケールは1
・編集ファイル２０４の全体のdurationは2秒
・追加データの全体のdurationは1秒
・各フラグメントのサイズは100kB
・各フラグメントのtfhdのbase_data_offsetは0

とする。

【０１４１】

上記のファイルに対して、図１８で示すように、編集ファイル２０４の先頭から１秒目にデータ(７０２)を追加する場合を説明する。

【０１４２】

この場合の、各編集単位のデータコピー以外のデータの変更内容は以下の点のみとなる。

（１）mehdのfragment_durationは、2から3に変更される

（２）フラグメント７０３trunのdata_offsetは、元の値+100に変更される。

【０１４３】

以上の編集内容が反映された形で出力ファイル２０６が生成される。

【０１４４】

＜編集の具体例３＞

さらに、moofとmoov、あるいはその逆のメタデータの再構成処理をともなう編集処理について説明する。まず、第３の例として、図１９のようにファイルの先頭部分をカットする場合の例を説明する。上述の編集の具体例１と同じ仕様を持つ編集ファイル２０４に対して、図１９で示すように、先頭から１秒（８０１)の部分をカットする場合を説明する。

【０１４５】

この例では、まず、各編集単位のデータコピー以外に以下のようなデータ変更が発生する。

（１）mehdのfragment_durationは、3から2に変更される

（２）フラグメント８０２、８０３のtrunのdata_offsetは、は元の値-100に変更される。

【０１４６】

さらに、フラグメント８０２は、編集後は出力ファイル２０６の先頭のフラグメントとなるため、上述の処理手順によってmoofからmoovへのメタデータ再構成処理が行われる。しかし、moovへの再構成処理が必要となるのはフラグメント８０２のみであるため、データ処理に与える影響は軽微なもので済む。

【０１４７】

＜編集の具体例４＞

第４の例として、図２０のようにファイルの先頭部分に別ファイルのデータを挿入する場合の例を説明する。上述の編集の具体例２と同じ仕様を持つ編集ファイル２０４に対して、図２０で示すように、先頭に１秒分のデータ（９０１)を挿入する場合を説明する。

【０１４８】

この例では、まず、各編集単位のデータコピー以外に以下のようなデータ変更が発生する。

（１）mehdのfragment_durationは、2から3に変更される

（２）フラグメント９０２、９０３のtrunのdata_offsetは、は元の値+100に変更される。

【０１４９】

さらに、フラグメント９０２は、編集後は出力ファイル２０６の２番目のフラグメントとなるため、上述の処理手順によってmoovからmoofへのメタデータ再構成処理が行われる。この場合も具体例３同様、moofへの再構成処理はフラグメント９０２に対してのみ行えば済むため、データ処理に与える影響はやはり軽微である。

【０１５０】

以上説明したように、上記の実施形態によれば、MP4ファイルの部分カットや追加などの編集におけるメタデータの変更は、ごくわずかなメタデータの編集で済むため、編集時のデータ処理負荷を縮小させることができる。

【０１５１】

（第２の実施の形態）

本発明の第１の実施形態では、処理を行うことができるファイルを識別するために、図

１０で示されるようなファイルの構成をチェックする処理が必要であった。しかし、処理可能なファイルの構成を識別する手段が別途提供される場合は、図１０のような手順でのチェック処理は必ずしも行う必要はない。そこで第２の実施の形態として、ファイル構成の識別手段が別途提供される場合を説明する。

【０１５２】

MP4ファイル形式は、UUIDをtypeに用いる形の拡張Boxを利用することや、あるいはUser Data Box(‘udta’)を利用することで、システムに固有の独自データを記録することが認められている。この仕組みを用いて、処理可能ファイルの識別に用いられるデータ構造を一意に識別する識別用データをファイル中に記述すれば、ファイル解析時に識別用データの有無をチェックするのみで済むため、処理対象ファイルのチェック処理を簡素化することが可能になる。

【０１５３】

また、上記識別用データとして、File Type Box(‘ftyp’)のブランド識別子を用いることも可能である。例えば処理可能なファイル形式を示すブランドを定義し、ファイルに記録することによって、ファイルの処理可否の判定はブランドのチェックのみで行えるようになる。

（その他の実施の形態）

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。

【０１５４】

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。

【０１５５】

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、光ディスク、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）などがある。

【０１５６】

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】

【０１５７】

【図１】メタデータ、メディアデータの概念を説明する図である。

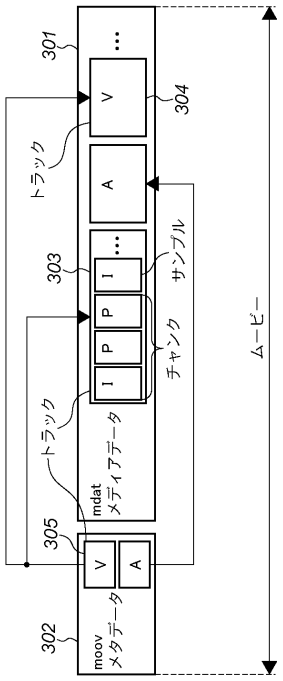
【図２】Boxのフィールド定義を示す図である。

【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】従来技術におけるMP4ファイルのカット編集処理を説明する図である。

【図５】本発明の実施形態におけるMP4ファイルのカット編集処理を説明する図である。
【図６】MPEG-4 VideoにおけるGOVの概念を説明する図である。
【図７】本発明の実施形態のデータ処理を実現する情報処理装置の構成を示すブロック図である。
【図８】本発明の実施形態のモジュール構成例を示す図である。
【図９】本発明の実施形態のカット編集処理の基本的な手順を説明するフローチャートである。
【図１０】本発明の実施形態において、ファイルが処理可能かチェックする処理を説明するフローチャートである。
【図１１】本発明の実施形態において、mdatの各トラックのメディアデータの構成を示す図である。
【図１２】フラグメントランの概念を説明する図である。
【図１３】本発明の実施形態において、moovをmoofに変換する処理を説明するフローチャートである。
【図１４】Track Extends Box(‘trex’)のフィールド定義を示す図である。
【図１５】本発明の実施形態のフラグメントの追加処理を説明するフローチャートである。
【図１６】本発明の実施形態のフラグメントの出力処理を説明するフローチャートである。
【図１７】本発明の実施形態のカット編集処理の例を示す図である。
【図１８】本発明の実施形態のデータ挿入処理の例を示す図である。
【図１９】本発明の実施形態の先頭部分の削除をともなうカット編集処理の例を示す図である。
【図２０】本発明の実施形態の先頭部分へのデータ挿入処理の例を示す図である。

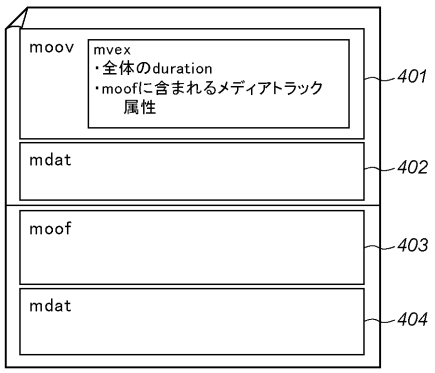
【図１】



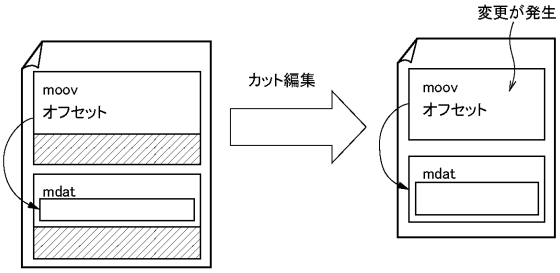
【図２】

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

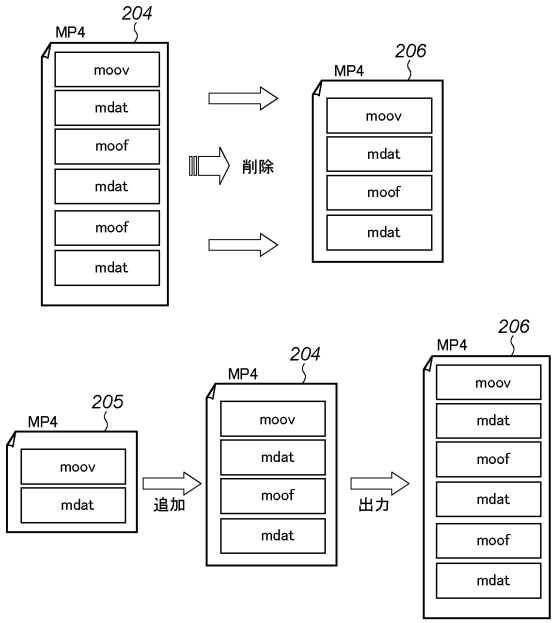
【図３】



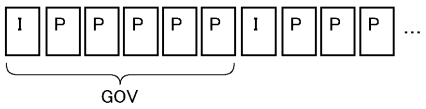
【図４】


moov
オフセット
mdat
カット編集
変更が発生
moov
オフセット
mdat

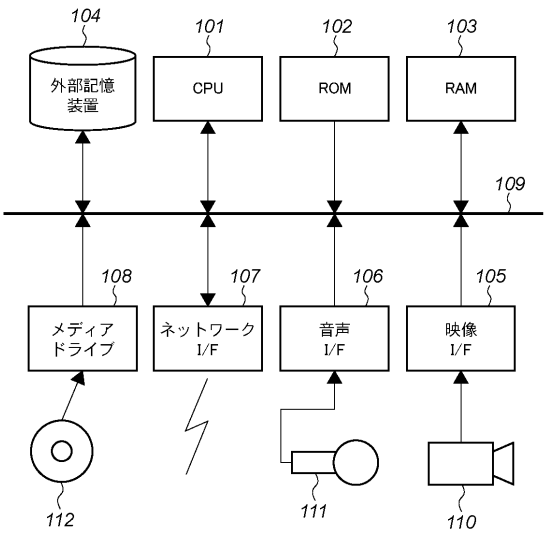
【図５】


204
MP4
moov
mdat
moof
mdat
moof
mdat
削除
206
MP4
moov
mdat
moof
mdat
205
MP4
moov
mdat
追加
204
MP4
moov
mdat
moof
mdat
出力
206
MP4
moov
mdat
moof
mdat
moof
mdat

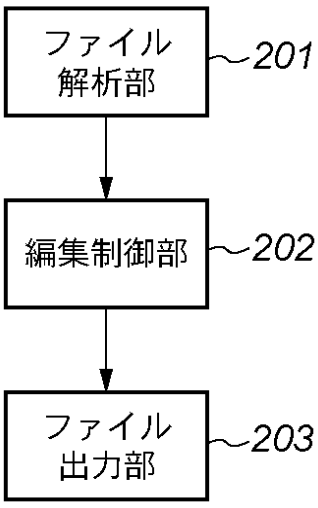
【図６】


I
P
P
P
P
P
I
P
P
P
...
GOV

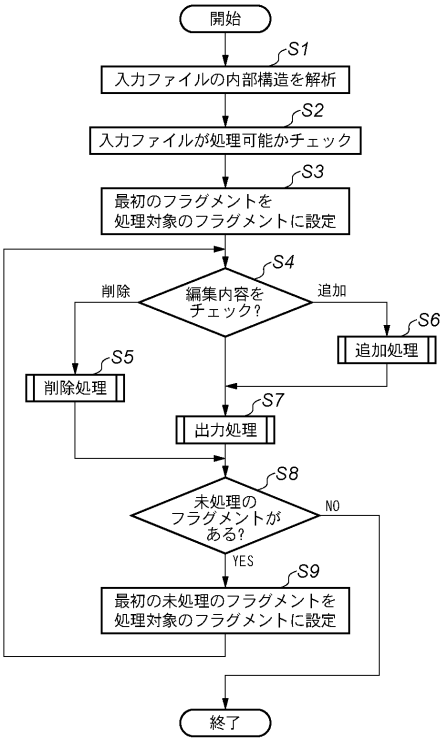
【図７】


104
外部記憶装置
101
CPU
102
ROM
103
RAM
109
108
メディアドライブ
107
ネットワークI/F
106
音声I/F
105
映像I/F
112
111
110

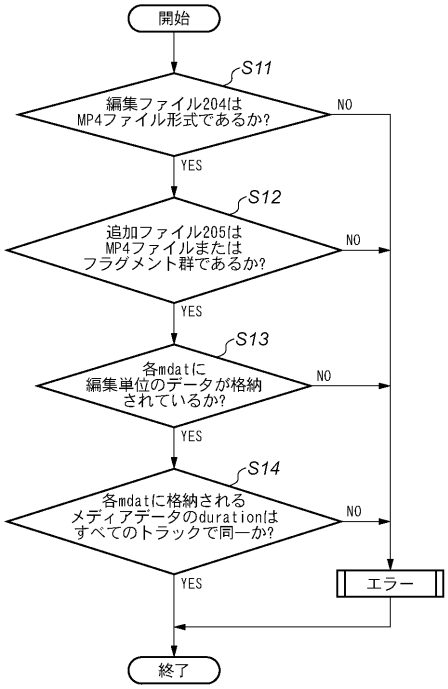
【図８】


ファイル解析部
201
編集制御部
202
ファイル出力部
203

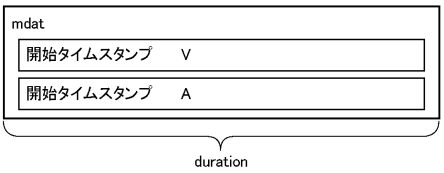
【図９】



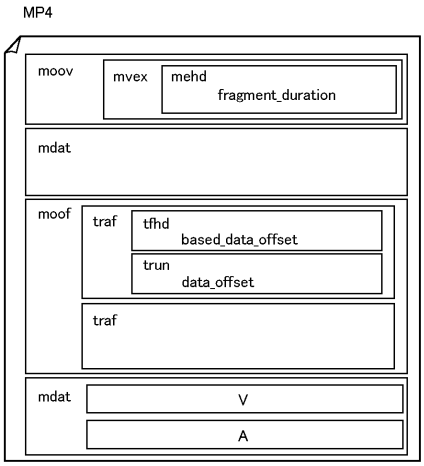
【図１０】



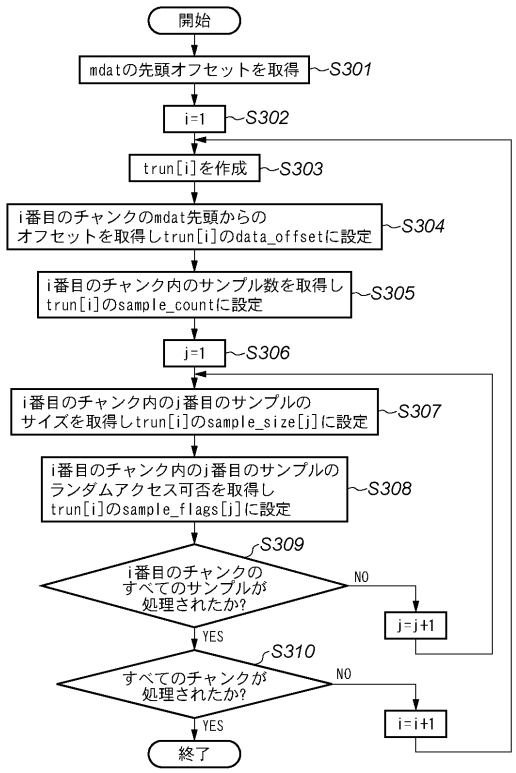
【図１１】



【図１２】



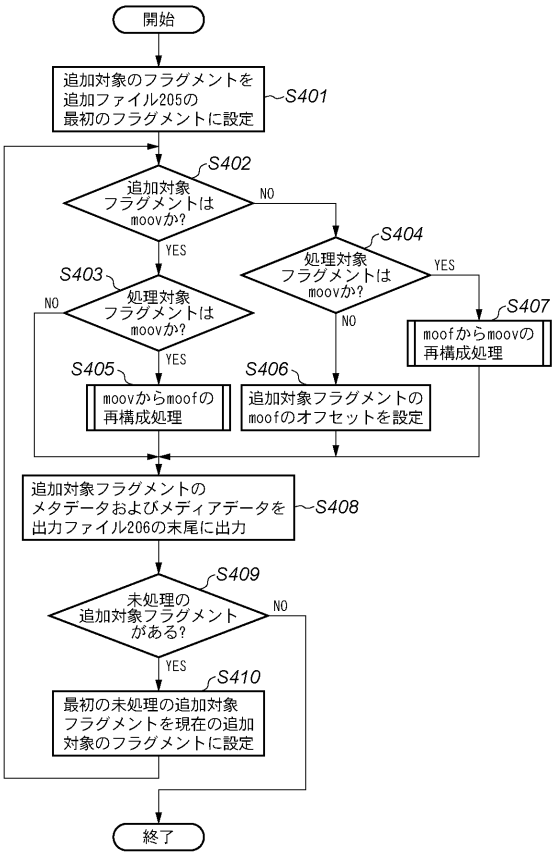
【図１３】



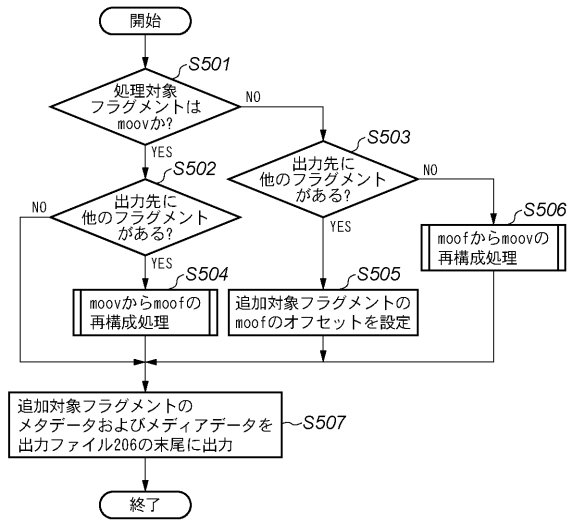
【図１４】

```
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 ;
}
```

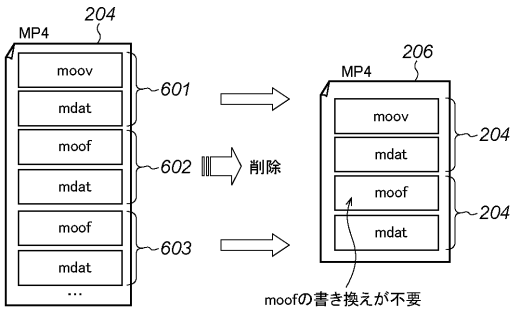
【図１５】



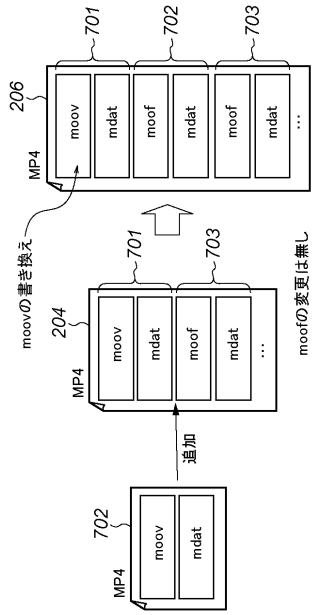
【図１６】



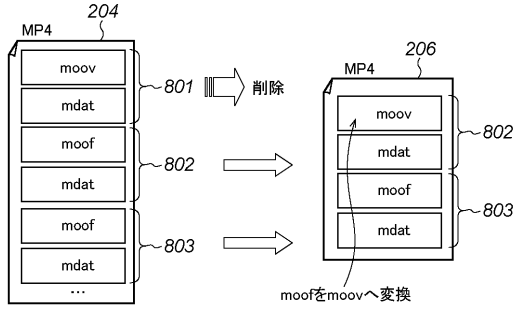
【図１７】



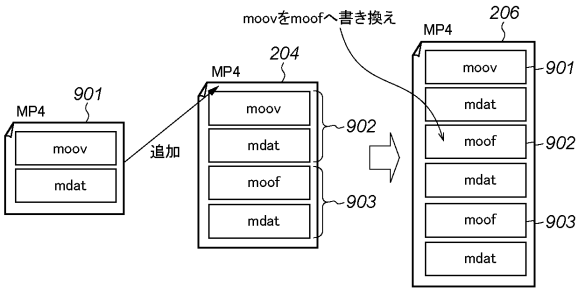
【図１８】


701
702
703
206
moov
mdat
moof
mdat
moof
mdat
…
MP4
moovの書き換え
701
703
204
moov
mdat
moof
mdat
…
MP4
moofの変更は無し
追加
702
moov
mdat
MP4

【図１９】


204
MP4
moov
mdat
moof
mdat
moof
mdat
…
801
802
803
削除
206
MP4
moov
mdat
moof
mdat
802
803
moofをmoovへ変換

【図２０】


moovをmoofへ書き換え
206
MP4
901
MP4
moov
mdat
追加
204
MP4
moov
mdat
moof
mdat
902
903
moov
mdat
moof
mdat
moof
mdat
901
902
903

フロントページの続き

(72)発明者　大嶋　肇
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
(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
(72)発明者　歌川　由香
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　中村　豊

(56)参考文献　特開２００４－００７６１０（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０４Ｎ　　５／９１