(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2010-250073
(P2010-250073A)
(43)公開日　平成22年11月4日(2010.11.4)

(51)Int.Cl.	F I	テーマコード（参考）
G10L 19/00 (2006.01)	G10L 19/00 220Z	

審査請求 未請求　請求項の数 9　ＯＬ　（全 19 頁）

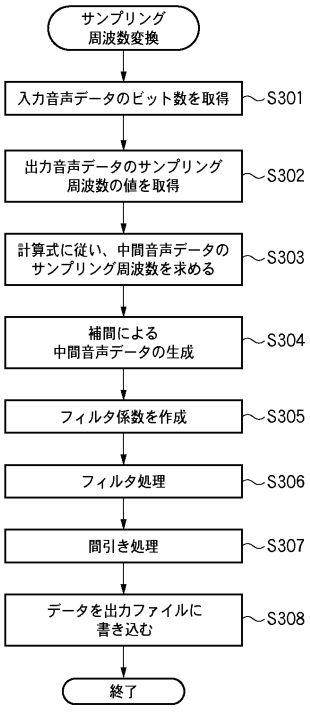
(21)出願番号　特願2009-99403(P2009-99403)
(22)出願日　平成21年4月15日(2009.4.15)

(71)出願人　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】音声処理装置および音声処理方法

(57)【要約】
【課題】音声データのサンプリング周波数変換処理を、出力音声データに求められる品質に応じて行う。
【解決手段】入力音声データのサンプリング周波数ｆｉを、サンプリング周波数変換後の出力音声データのサンプリング周波数ｆｏの整数倍のサンプリング周波数ｆｐに、補間処理を用いて変換する。このとき、入力音声データの量子化ビット数の関数を用いて、周波数ｆｐが量子化ビット数が小さいときは低い周波数に、量子化ビット数が大きいときは高い周波数になるように、周波数ｆｏに対する倍率を求め、この倍率と周波数ｆｏとから周波数ｆｐを求める。その後、サンプリング周波数ｆｐの中間音声データに対して間引き処理を行い、サンプリング周波数ｆｏの出力音声データを得る。量子化ビット数が大きいときは、高品質の出力音声データが得られ、量子化ビット数が小さいときは、補間処理の負荷が低減される。
【選択図】図４



【特許請求の範囲】
【請求項１】
音声データの第１のサンプリング周波数を、該第１のサンプリング周波数とは異なる第２のサンプリング周波数に変換する音声処理装置であって、
前記音声データの量子化ビット数と、前記第１のサンプリング周波数と、前記第２のサンプリング周波数とを取得する取得手段と、
前記量子化ビット数および前記第２のサンプリング周波数とに基づき、該第２のサンプリング周波数の整数倍の第３のサンプリング周波数を求める算出手段と、
前記音声データに対してサンプルの補間処理を行うことで前記第１のサンプリング周波数を前記第３のサンプリング周波数に変換する第１の変換手段と、
前記第１の変換手段でサンプリング周波数が前記第３のサンプリング周波数に変換された前記音声データに対して、前記整数倍に応じたサンプルの間引き処理を行うことで、前記第３のサンプリング周波数を前記第２のサンプリング周波数に変換する第２の変換手段と
を有し、
前記算出手段は、前記第３のサンプリング周波数として、前記量子化ビット数が大きいほど高い周波数を求め、前記量子化ビット数が小さいほど低い周波数を求める
ことを特徴とする音声処理装置。
【請求項２】
前記算出手段は、
前記量子化ビット数の関数を用いて前記第２のサンプリング周波数を前記第３のサンプリング周波数に変換する際の倍率を算出し、該倍率を前記第２のサンプリング周波数に乗じることで前記第３のサンプリング周波数を算出する
ことを特徴とする請求項１に記載の音声処理装置。
【請求項３】
前記関数は、
前記第２の変換手段で前記第２のサンプリング周波数に変換された前記音声データのＳＮ比が所定以上になる前記第３のサンプリング周波数に基づき決定する
ことを特徴とする請求項２に記載の音声処理装置。
【請求項４】
前記算出手段は、
前記量子化ビット数と、前記第２のサンプリング周波数を前記第３のサンプリング周波数に変換する際の倍率とを関連付けたテーブルを用いて前記第３のサンプリング周波数を求める
ことを特徴とする請求項１に記載の音声処理装置。
【請求項５】
前記テーブルは、
前記量子化ビット数の範囲と前記倍率とが関連付けられる
ことを特徴とする請求項４に記載の音声処理装置。
【請求項６】
前記テーブルは、
前記第２の変換手段で前記第２のサンプリング周波数に変換された前記音声データのＳＮ比が所定以上になる前記第３のサンプリング周波数に基づき作成する
ことを特徴とする請求項４または請求項５に記載の音声処理装置。
【請求項７】
前記テーブルの前記倍率および前記量子化ビット数は、ユーザ操作により設定が可能である
ことを特徴とする請求項４または請求項５に記載の音声処理装置。
【請求項８】
音声データの第１のサンプリング周波数を、該第１のサンプリング周波数とは異なる第

２のサンプリング周波数に変換する音声処理方法であって、
取得手段が、前記音声データの量子化ビット数と、前記第１のサンプリング周波数と、前記第２のサンプリング周波数とを取得する取得ステップと、
算出手段が、前記量子化ビット数および前記第２のサンプリング周波数とに基づき、該第２のサンプリング周波数の整数倍の第３のサンプリング周波数を求める算出ステップと、
第１の変換手段が、前記音声データに対してサンプルの補間処理を行うことで前記第１のサンプリング周波数を前記第３のサンプリング周波数に変換する第１の変換ステップと、
第２の変換手段が、前記第１の変換ステップでサンプリング周波数が前記第３のサンプリング周波数に変換された前記音声データに対して、前記整数倍に応じたサンプルの間引き処理を行うことで、前記第３のサンプリング周波数を前記第２のサンプリング周波数に変換する第２の変換ステップと
を有し、
前記算出ステップでは、前記第３のサンプリング周波数として、前記量子化ビット数が大きいほど高い周波数を求め、前記量子化ビット数が小さいほど低い周波数を求める
ことを特徴とする音声処理方法。
【請求項９】
コンピュータを請求項１乃至請求項７の何れか１項に記載の音声処理装置の各手段として機能させるためのプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、音声データのサンプリング周波数を変換する音声処理装置および音声処理方法に関する。
【背景技術】
【０００２】
音声データを処理する機器で使用される音声データの規格は、サンプリング周波数がＣＤ－ＤＡで４４．１ｋＨｚ、ＤＶＤ－Ａｕｄｉｏで９６ｋＨｚ、また、ＤＶで４８ｋＨｚと、機器により異なっている。そのため、異なった規格の機器間で音声データをやりとりする場合、音声データのサンプリング周波数を変換する必要がある。
【０００３】
マルチレートのサンプリング周波数変換の手法として、非特許文献１に記載されるような手法が従来から知られている。非特許文献１によれば、入力音声データおよび出力音声データそれぞれのサンプリング周波数の最小公倍数を求める。そして、入力音声データのサンプリング周波数を、最小公倍数で示される値の中間サンプリング周波数に変換し、中間音声データを得る。この中間音声データに対して低域通過フィルタを用いた間引き処理を施して、所望のサンプリング周波数の出力音声データを得る。
【先行技術文献】
【非特許文献】
【０００４】
【非特許文献１】電子情報通信学会誌，１９８７年１０月，ｐ.１０３８－１０４１
【発明の概要】
【発明が解決しようとする課題】
【０００５】
入力音声データのサンプリング周波数が４４．１ｋＨｚ、出力音声データのサンプリング周波数が４８ｋＨｚの場合、入力音声データおよび出力音声データのサンプリング周波数の最小公倍数は、７．０５６ＭＨｚとなる。このとき、入力音声データに対してサンプリング周波数を１６０倍のサンプリング周波数に変換して中間音声データを生成し、この中間音声データに対して低域通過フィルタ処理を施して１／１４７倍に間引くため、処理

に時間がかかるという問題点があった。つまり、入力音声データおよび出力音声データのサンプリング周波数が簡単な整数比で表されない場合、処理に時間がかかることになる。

【０００６】

また、従来では、入力音声データのビット数を考慮せずに、入力音声データのサンプリング周波数の中間サンプリング周波数への変換を行っていた。より具体的には、従来では、入力音声データの量子化ビット数が比較的低い場合（８ビット、１２ビットなど）でも、量子化ビット数が高い場合（１６ビット、２４ビットなど）と同様の処理を行っていた。そのため、処理時間が必要以上にかかってしまうという問題点があった。

【０００７】

すなわち、入力音声データの量子化ビット数が高い場合、出力音声データに対し、より高品質が求められていると考えられる。一方、入力音声データの量子化ビット数が比較的低い場合は、出力音声データに対して高品質が求められていないことが考えられる。このように、出力音声データに対して高品質が求められていない場合に、出力音声データに対して高品質が求められている場合と同じだけの処理時間をかけるのは、無駄である。

【０００８】

したがって、本発明の目的は、音声データのサンプリング周波数変換処理を、出力音声データに求められる品質に応じて行うことができる音声処理装置および音声処理方法を提供することにある。

【課題を解決するための手段】

【０００９】

本発明は、上述した課題を解決するために、音声データの第１のサンプリング周波数を、第１のサンプリング周波数とは異なる第２のサンプリング周波数に変換する音声処理装置であって、音声データの量子化ビット数と、第１のサンプリング周波数と、第２のサンプリング周波数とを取得する取得手段と、量子化ビット数および第２のサンプリング周波数とに基づき、第２のサンプリング周波数の整数倍の第３のサンプリング周波数を求める算出手段と、音声データに対してサンプルの補間処理を行うことで第１のサンプリング周波数を第３のサンプリング周波数に変換する第１の変換手段と、第１の変換手段でサンプリング周波数が第３のサンプリング周波数に変換された音声データに対して、整数倍に応じたサンプルの間引き処理を行うことで、第３のサンプリング周波数を第２のサンプリング周波数に変換する第２の変換手段とを有し、算出手段は、第３のサンプリング周波数として、量子化ビット数が大きいほど高い周波数を求め、量子化ビット数が小さいほど低い周波数を求めることを特徴とする音声処理装置である。

【発明の効果】

【００１０】

本発明は、音声データのサンプリング周波数変換処理を、出力音声データに求められる品質に応じて行うことができる。

【図面の簡単な説明】

【００１１】

【図１】本発明による音声処理装置として適用可能な情報処理装置の一例の構成を示すブロック図である。

【図２】ＷＡＶファイルの一例の構造を示す図である。

【図３】ユーザが入力音声データのサンプリング周波数変換を情報処理装置に指示するための一例のユーザインターフェイスを示す図である。

【図４】本発明の第１の実施形態によるサンプリング周波数変換の一例の処理を示すフローチャートである。

【図５】本発明の第１の実施形態に適用可能なフィルタ係数の一例の算出処理について説明するためのフローチャートである。

【図６】本発明の第１の実施形態によるサンプリング周波数の変換処理を説明するための図である。

【図７】本発明の第１の実施形態によるサンプリング周波数の変換処理を説明するための

図である。
【図８】本発明の第１の実施形態によるサンプリング周波数の変換処理を説明するための図である。
【図９】本発明の第１の実施形態によるサンプリング周波数の変換処理を説明するための図である。
【図１０】本発明の第１の実施形態によるサンプリング周波数の変換処理を説明するための図である。
【図１１】本発明の第２の実施形態によるサンプリング周波数変換の一例の処理を示すフローチャートである。
【発明を実施するための形態】
【００１２】
＜第１の実施形態＞
以下、本発明の第１の実施形態を、図面を参照しながら説明する。図１は、本発明による音声処理装置として適用可能な情報処理装置１００の一例の構成を示す。図１において、バス１０１に対してＣＰＵ１１０、ＲＡＭ１１１およびＲＯＭ１１２、ならびに、グラフィクスコントローラ１１３が接続される。バス１０１に対して、さらに、入力インターフェイス（Ｉ／Ｆ）１２０、ハードディスク１２２、ドライブ装置１２３およびオーディオインターフェイス（Ｉ／Ｆ）１２４が接続される。情報処理装置１００の各部は、バス１０１を介してデータのやりとりを行うことができる。
【００１３】
ＣＰＵ１１０は、ＲＯＭ１１２に予め記憶されたプログラムに基づきシステムを起動させる。そして、ＣＰＵ１１０は、後述するハードディスク１２２に予め記録されるプログラムデータを読み込んでＲＡＭ１１１に展開し、ＲＡＭ１１１上に展開されたプログラムデータに基づき情報処理装置１００の各部を制御する。
【００１４】
グラフィクスコントローラ１１３は、表示装置１１４が接続され、ＣＰＵ１１０から出力された表示制御信号を表示装置１１４が表示可能な信号に変換して、表示装置１１４に出力する。表示装置１１４は、例えばＬＣＤ(Liquid Crystal Display)を表示素子として用いる。
【００１５】
入力インターフェイス１２０は、ユーザ操作を受け付けるための入力デバイス、例えばキーボード１２１Ａと、ポインティングデバイス１２１Ｂとが接続される。ポインティングデバイス１２１Ｂは、マウスやタッチパッドを適用できる。キーボード１２１Ａやポインティングデバイス１２１Ｂから出力された信号は、入力インターフェイス１２０を介してＣＰＵ１１０に渡される。
【００１６】
ハードディスク１２２は、記録媒体そのものであるハードディスク本体と、ハードディスクを駆動するためのドライブ装置とからなるハードディスクドライブ（ＨＤＤ）として用いられる。ハードディスク１２２には、上述した、ＣＰＵ１１０を動作させるためのプログラムが記録されると共に、この情報処理装置１００で生成されたり、この情報処理装置１００に入力される様々なデータを格納することができる。
【００１７】
ドライブ装置１２３は、記録可能なタイプのＤＶＤやＣＤといった記録媒体が装填可能とされ、装填された記録媒体に対するデータの読み書きを行う。勿論、ドライブ装置１２３に対して、再生専用のタイプのＤＶＤやＣＤ－ＲＯＭを装填してもよい。これに限らず、ドライブ装置１２３は、書き換え可能な不揮発性メモリを装着可能とし、この不揮発性メモリに対するデータの読み書きを行うようにしてもよい。さらに、ドライブ装置１２３は、フレキシブルディスクに対応したものでもよい。
【００１８】
通信Ｉ／Ｆ１２４は、所定のプロトコルを用いて外部の機器やネットワークに接続する

。例えば、通信Ｉ／Ｆ１２４は、ＵＳＢやＩＥＥＥ１３９４といったデータ転送プロトコルに対応し、外部機器から転送されたデータを受信すると共に、外部機器に対してデータの転送を行う。これに限らず、通信Ｉ／Ｆ１２４は、ＴＣＰ／ＩＰに対応し、インターネットやＬＡＮ(Local Area Network)との接続を制御するようなものでもよい。この通信Ｉ／Ｆ１２４を介して、この情報処理装置１００における音声データの入出力を行うことができる。

【００１９】

次に、本発明に適用可能な音声データのフォーマットについて、概略的に説明する。以下では、以下では、音声データがＰＣＭ方式で符号化され、ＷＡＶファイル形式でファイルに格納されるものとする。なお、ＰＣＭは、Pulse Code Modulationの略称である。

【００２０】

図２は、ＷＡＶファイルの一例の構造を示す。領域４０１は、４バイトのデータ長を有し、このファイルがＲＩＦＦ形式であることを識別する文字列「ＲＩＦＦ」が格納される。なお、ＲＩＦＦは、Resource Interchange File Formatの略称である。領域４０２は、４バイトのデータ長を有し、このファイルの領域４０２以降のファイルサイズがバイト単位で格納される。すなわち、領域４０２に格納されるファイルサイズは、このＷＡＶファイル全体のファイルサイズよりも８バイト分少ないことになる。

【００２１】

領域４０３は、４バイトのデータ長を有し、ＲＩＦＦの種類を表す識別子が格納される。この例では、文字列「ＷＡＶＥ」が格納され、このファイルがＷＡＶファイル形式で音声データを格納するファイルであることが示される。領域４０４は、４バイトのデータ長を有し、フォーマットを定義するためのｆｍｔチャンクが格納される領域を示す文字列「ｆｍｔ　」（空白文字を含めた４文字）が格納される。領域４０５は、４バイトのデータ長を有し、ｆｍｔチャンクによる音声データの特性を示す情報のデータサイズが格納される。

【００２２】

領域４０６は、ｆｍｔチャンクの本体、すなわち、このＷＡＶファイルに格納される音声データの特性を示す情報が格納される。例えば、領域４０６には、サンプリング周波数や、サンプル当たりの量子化ビット数などの情報が格納される。なお、サンプリング周波数は、Ｈｚ（ヘルツ）単位で記述されて領域４０６に格納される。

【００２３】

領域４０７は、４バイトのデータ長を有し、ＰＣＭ方式の音声データの波形データが格納される領域を示す文字列「ＤＡＴＡ」が格納される。領域４０８は、次の領域４０９に格納される波形データのデータサイズが格納される。領域４０９には、ＰＣＭ方式で符号化された音声の波形データが格納される。領域４０９に格納される波形データが、このＷＡＶファイルにおける音声データの本体となる。

【００２４】

なお、ここでは音声データがＰＣＭ方式で変調および符号化され、ＷＡＶファイルに格納されるものとして説明したが、これはこの例に限定されない。すなわち、本発明は、サンプル当たりの量子化ビット数およびサンプリング周波数の値を取得できれば、ＡＩＦＦ形式など他の形式の音声データにも適用可能である。なお、ＡＩＦＦは、Audio Interchange File Formatの略称である。

【００２５】

図３は、ユーザが入力音声データのサンプリング周波数変換を情報処理装置１００に指示するための一例のユーザインターフェイスを示す。このユーザインターフェイスは、ＣＰＵ１１０によりプログラムに従い生成され、例えば表示装置１１４に対して表示される。

【００２６】

画面２０１に対して、入力音声データファイルのパスを表示するパス表示部２０２と、入力音声データファイルのパスを参照するための参照ボタン２０３と、入力音声データの

サンプリング周波数を表示する表示部２０４とが表示される。さらに、画面２０１に対して、出力音声データのパスを表示するパス表示部２０５と、出力音声データファイルのパスを参照するための参照ボタン２０６と、出力音声データのサンプリング周波数を指定するための入力部２０７とが表示される。また、変換ボタン２０８は、画面２０１で設定された内容に従い、入力音声データのサンプリング周波数の変換処理を開始させるためのものである。

【００２７】

ユーザにより、画面２０１の表示に従いキーボード１２１Ａやポインティングデバイス１２１Ｂなどの入力デバイスが操作され、入力音声データのサンプリング周波数を変換するための情報が情報処理装置１００に対して入力される。

【００２８】

一例として、ユーザ操作により参照ボタン２０３が操作されると、ＣＰＵ１１０は、例えばハードディスク１２２に格納される音声データファイルから変換を行いたいファイルを指定するためのダイアログ（図示しない）を表示させる。このダイアログに従い、ユーザ操作によりサンプリング周波数を変換する入力音声データファイルが指定されると、ＣＰＵ１１０は、指定された入力音声データファイルのパスをパス表示部２０２に表示させる。また、ＣＰＵ１１０は、指定された入力音声データファイルの領域４０６から当該ファイルに格納される音声データのサンプリング周波数を示す情報を取得し、表示部２０４に対して例えばヘルツ単位で表示させる。

【００２９】

ユーザ操作により参照ボタン２０６が操作されると、ＣＰＵ１１０は、サンプリング周波数を変換された出力音声データファイルを保存するためのフォルダを指定するダイアログ（図示しない）を表示させる。このダイアログに従い、ユーザ操作によりフォルダが指定され、さらにファイル名が指定されると、ＣＰＵ１１０は、指定された出力音声データファイルへのパスをパス表示部２０５に表示させる。さらに、ユーザ操作により変換後のサンプリング周波数の値が入力部２０７に対して入力されると、入力された値がＲＡＭ１１１やＣＰＵ１１０のレジスタなどに保持される。なお、ユーザ操作により出力音声データファイルのパスをパス表示部２０５に対して直接的に入力することも可能である。また、変換後のサンプリング周波数の値は、予め決められた幾つかの値から選択するようにもできる。

【００３０】

上述のようにして、画面２０１に従い、入力音声データに対するサンプリング周波数変換に必要な情報が指定された後に、ユーザ操作により変換ボタン２０８が操作されると、ＣＰＵ１１０は、本第１の実施形態によるサンプリング周波数の変換処理を開始する。すなわち、パス表示部２０２に表示された入力音声データファイルに格納された入力音声データに対して、当該入力音声データのサンプリング周波数を入力２０７に入力されたサンプリング周波数へと変換する処理が開始される。この処理によりサンプリング周波数が変換された音声データは、ファイルに格納され、出力音声データファイルとして、パス表示部２０５に表示されるパスに保存される。

【００３１】

図４は、本第１の実施形態によるサンプリング周波数変換の一例の処理を示すフローチャートである。ここでは、サンプリング周波数が１１．０２５ｋＨｚ、量子化ビット数が１６ビットの入力音声データについて、サンプリング周波数を異なる周波数、例えば１２ｋＨｚに変換する例を用いて説明する。入力音声データのサンプリング周波数である１１．０２５ｋＨｚと、出力音声データのサンプリング周波数である１２ｋＨｚは、簡単な整数比の関係に無い。

【００３２】

最初のステップＳ３０１で、ＣＰＵ１１０は、サンプリング周波数の変換対象である入力音声データの量子化ビット数を取得する。具体的には、ＣＰＵ１１０は、パス表示部２０２に表示される音声データファイルの領域４０６にアクセスして、当該音声データファ

イルに格納される音声データの量子化ビット数の情報を取得する。この例では、入力音声データの量子化ビット数として１６ビットが取得される。

【００３３】

次のステップＳ３０２で、ＣＰＵ１１０は、入力部２０７に入力された、変換後のサンプリング周波数ｆｏ（第２のサンプリング周波数）を取得する。この例では、入力部２０７に入力された１２０００Ｈｚ（１２ｋＨｚ）が出力音声データのサンプリング周波数ｆｏとして取得される。このサンプリング周波数ｆｏが、出力音声データのサンプリング周波数となる。

【００３４】

入力音声データの量子化ビット数と、出力音声データのサンプリング周波数ｆｏとが取得されると、処理はステップＳ３０３に移行される。ステップＳ３０３で、ＣＰＵ１１０は、取得された入力音声データの量子化ビット数と、出力音声データのサンプリング周波数ｆｏとに基づき、中間音声データのサンプリング周波数ｆｐ（第３のサンプリング周波数）を求める。

【００３５】

すなわち、入力音声データのサンプリング周波数ｆｉ（第１のサンプリング周波数）を、出力音声データのサンプリング周波数ｆｏの整数倍のサンプリング周波数ｆｐに変換（第１の変換）して中間音声データを生成する。そして、この中間音声データのサンプルを所定に間引いて（第２の変換）、サンプリング周波数ｆｏの出力音声データを生成する。

【００３６】

本発明では、中間音声データのサンプリング周波数ｆｐを、入力音声データの量子化ビット数が低いほど低くし、入力音声データの量子化ビット数が高いほど高くする。このようにすることで、入力音声データの量子化ビット数が低い場合には、後述する低域通過フィルタ処理のＣＰＵに対する負荷を、入力音声データの量子化ビット数が高い場合よりも低減することが可能となる。また、入力音声データの量子化ビット数が高い場合には、出力音声データにおいて高品質を達成することが可能となる。

【００３７】

本第１の実施形態では、この入力音声データの量子化ビット数と中間音声データのサンプリング周波数ｆｐとの関係を達成するために、中間音声データのサンプリング周波数ｆｐを、入力音声データの量子化ビット数の関数として求める。この関数の一例を下記の式（１）に示す。なお、式（１）において、値ｘは、入力音声データの量子化ビット数を表す。値ｙは、中間音声データのサンプリング周波数ｆｐの、出力音声データのサンプリング周波数ｆｏに対する倍率を表す。また、値ｃは、所定の定数である。

$$y = cx \quad \cdots (1)$$

【００３８】

なお、定数ｃは、例えば、ソフトウェアあるいは装置が対応可能な最高量子化ビット数において所定以上のＳＮ比を達成可能な最も低い中間音声データのサンプリング周波数ｆｐを実験的に求め、このサンプリング周波数ｆｐに基づき決定する。この定数ｃの算出方法の具体的な例については、後述する。

【００３９】

定数ｃの値をｃ＝０．３７５とした場合、入力音声データの量子化ビット数が１６ビットであるので、式（１）によりｙ＝６となり、中間音声データのサンプリング周波数ｆｐ＝６×１２ｋＨｚ＝７２ｋＨｚとなる。

【００４０】

ステップＳ３０３で中間音声データのサンプリング周波数ｆｐが算出されると、処理は次のステップＳ３０４に移行され、ＣＰＵ１１０により中間音声データの生成が行われる。より具体的には、ステップＳ３０４では、ステップＳ３０３で算出された中間音声データのサンプリング周波数ｆｐ（この例では７２ｋＨｚ）になるように、補間処理によりサンプルを生成して中間音声データを生成する。

【００４１】

本第１の実施形態では、この補間処理を、ラグランジュ補間の二次多項式補間を用いて行う。例えば、３点(x_1,y_1)、(x_2,y_2)および(x_3,y_3)から点Ｐ(ｘ,ｙ)のｙ座標を補間により求める場合、下記の式（２）を用いる。

【数１】

$$y=\frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}y_1+\frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}y_2+\frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}y_3 \quad \cdots (2)$$

【００４２】

次のステップＳ３０５で、ＣＰＵ１１０は、低域通過フィルタのフィルタ係数を生成する。すなわち、ステップＳ３０５では、上述したステップＳ３０４の補間により発生した余分な高周波成分を除去するために用いる低域通過フィルタのフィルタ係数を生成する。

【００４３】

図５のフローチャートを用いて、ステップＳ３０５におけるフィルタ係数の一例の算出処理について説明する。なお、図５のフローチャートの各処理は、ＣＰＵ１１０によりプログラムに従い実行される。

【００４４】

最初のステップＳ５０１で、ＣＰＵ１１０は、低域通過フィルタのカットオフ周波数ｆｃを求める。入力音声データおよび出力音声データのサンプリング周波数のうち、周波数が低い方のサンプリング周波数の１／２の値をカットオフ周波数ｆｃとする。この例では、入力音声データのサンプリング周波数ｆｉ（１１．０２５ｋＨｚ）の方が出力音声データのサンプリング周波数ｆｏ（１２ｋＨｚ）よりも低い。したがって、入力音声データのサンプリング周波数ｆｉである１１．０２５ｋＨｚの１／２の５．５１２５ｋＨｚをカットオフ周波数ｆｃとする。

【００４５】

そして、次のステップＳ５０２で、カットオフ周波数ｆｃを、低域通過フィルタに入力する中間音声データのサンプリング周波数ｆｐで規格化する。この例では、ステップＳ５０２において、カットオフ周波数ｆｃを中間音声データのサンプリング周波数ｆｐの７２ｋＨｚで除して、ｆｃ＝０．０７６４６２５に規格化する。

【００４６】

ステップＳ５０３で、フィルタの長さＮを求める。理想的な低域通過フィルタは係数が無限に続くので、係数を有限にするために、窓関数を使用してフィルタの長さを打ち切ることが必要となる。窓関数によって有限の係数になったときのフィルタの係数の個数をフィルタの長さＮと呼ぶことにする。本第１の実施形態では、窓関数としてハニング窓を使用する。ハニング窓を使用する場合、窓の長さN_wは、以下の式（３）を用いて算出する。なお、算出結果の小数部分は、切り上げる。また、計算の結果、窓の長さN_wの値が偶数になった場合は、１を加えて奇数にする。

【数２】

$$N_w=\frac{3.1}{0.2f_c} \quad \cdots (3)$$

【００４７】

次のステップＳ５０４では、下記の式（４）を用いてフィルタ係数ｈ（ｎ）を算出する。

【数３】

$$h(n)=\frac{\sin 2\pi f_c\left(n-\frac{N-1}{2}\right)}{2\pi f_c\left(n-\frac{N-1}{2}\right)}2f_c w(n) \qquad (n=0,\mathrm{K},N-1)$$

… (4)

$$h(n)=2f_c w(n) \qquad \left(n=\frac{N-1}{2}\right)$$

【００４８】

なお、式（４）において関数ｗ（ｘ）は窓関数で、ハニング窓を使用する場合、下記の式（５）で表される。但し、０≦ｘ≦１である。

【数４】

$$w(x)=0.5-0.5\cos 2\pi x \qquad \cdots (5)$$

【００４９】

説明は図４のフローチャートに戻り、ステップＳ３０５でフィルタ係数ｈ（ｎ）が算出されると、処理はステップＳ３０６に移行される。ステップＳ３０６で、ＣＰＵ１１０は、ステップＳ３０５で生成されたフィルタ係数ｈ（ｎ）を適用した低域通過フィルタにより、中間音声データに対してフィルタ処理を行う。ここでは、ステップＳ３０５で作成したフィルタ係数を下記の式（６）の通り使用して、カットオフ周波数ｆｃよりも大きな周波数成分が除去された中間音声データを求める。

【数５】

$$y(n)=\sum_{k=0}^{N-1}h(k)x(n-k) \qquad \cdots (6)$$

【００５０】

ステップＳ３０６で中間音声データに対するフィルタ処理が行われると、処理はステップＳ３０７に移行され、ステップＳ３０６で得られた中間音声データを１／Ｎに間引き、出力音声データを生成する。この出力音声データは、次のステップＳ３０８で、出力パスで指定されたファイルに書き込まれる。

【００５１】

次に、サンプリング周波数ｆｉ＝１１．０２５ｋＨｚ、量子化ビット数が１６ビットの入力音声データを、サンプリング周波数ｆｏ＝１２ｋＨｚ、量子化ビット数が１６ビットの出力音声データに変換する処理について、より具体的な例を用いて説明する。

【００５２】

なお、以下に示す図６（ａ）～図６（ｄ）において、横軸は時間を示し、縦軸はサンプルのレベル（ビット値）を示す。また、図７（ａ）～図７（ｄ）において、横軸は周波数を示し、縦軸はエネルギを示す。

【００５３】

図６（ａ）は、サンプリング周波数ｆｉが１１．０２５ｋＨｚの入力音声データの例を示す。図６（ａ）の入力音声データは、図７（ａ）のスペクトル９０１に例示されるように、スペクトルが周波数ｆｉ＝１１．０２５ｋＨｚの周期で繰り返されるものとなる。図６（ａ）のサンプルあたりの間隔Ｔｉは１／１１．０２５ｋＨｚ　０．０９０７（ミリ秒）である。

【００５４】

図４のステップＳ３０４で、サンプリング周波数ｆｐが出力音声データのサンプリング周波数ｆｏ（１２ｋＨｚ）のＮ倍である中間音声データを補間処理により生成する。補間処理により生成された中間音声データの例を図６（ｂ）に、そのスペクトルの例を図７（ｂ）に示す。スペクトル９０２は、補間処理によって生じた雑音である。補間を行う際の

次数が高くなると補間の精度が高くなるため、雑音のスペクトル９０２は小さくなる。また、フィルタの長さＮの値を大きくすると補間の精度が高くなるため、雑音のスペクトル９０２は小さくなる。図６（ｂ）に例示される中間音声データのサンプル間隔Ｔｐは、１／(Ｔｏ×Ｎ)＝１／(１２ｋＨｚ×Ｎ)　０．０８３３／Ｎ（ミリ秒）である。

【００５５】

図４のステップＳ３０６で、中間音声データに対して低域通過フィルタ処理が施され、補間処理によって発生した高周波成分を除去する。ここで用いる低域通過フィルタは、カットオフ周波数ｆｃが入力音声データのサンプリング周波数ｆｉ＝１１．０２５ｋＨｚの１／２の５．５０１２５ｋＨｚとなっている。高周波成分を除去されたデータの例を図６（ｃ）に、そのスペクトルの例を図７（ｃ）に示す。

【００５６】

図４のステップＳ３０７で、中間音声データのサンプルを１／Ｎに間引いて出力音声データを生成する。出力音声データの例を図６（ｄ）に、そのスペクトルの例を図７（ｄ）に示す。図６（ｄ）に例示される出力音声データのサンプル間隔Ｔｏは１／１２（ｋＨｚ）＝０．０８３３（ミリ秒）である。

【００５７】

＜第１の実施形態による具体的な処理結果の例＞

量子化ビット数が１６ビットである入力音声データのサンプリング周波数ｆｉを１１．０２５ｋＨｚから１２ｋＨｚに変換する処理における、処理時間の実測値の例を図８に示し、出力音声データのＳＮ比の実測値の例を図９に示す。なお、図９の縦軸はステップＳ３０６のフィルタ処理における処理時間を示し、図１０の縦軸は出力音声データのＳＮ比を示す。また、図９および図１０の横軸は、補間処理によって生成された中間音声データのサンプリング周波数ｆｐ（単位：ｋＨｚ）を示す。

【００５８】

図９および図１０の各図において、補間処理にラグランジュの一次多項式補間を用いた例（破線）とラグランジュの二次多項式補間を用いた例（実線）とをそれぞれ示す。この例では、入力音声データとして、量子化ビット数が１６ビット、再生時間長が３０秒のＰＣＭ形式の音声データを使用している。

【００５９】

フィルタの長さＮの値を大きくすると、入力音声データのサンプル間が細かく補間され（図８(ｂ)参照）、補間処理によって発生する周波数成分が高い周波数に分散されることが分かる（図９(ｂ)参照）。高周波成分は、低域通過フィルタで除去すればよい。

【００６０】

一次多項式補間よりも二次多項式補間の方が、精度の高い補間ができる。そのため、二次多項式補間では、補間によって発生する高周波成分は一次多項式補間よりも少ない。図９では、二次多項式補間の方が一次多項式補間よりもＳＮ比が高いことが示されている。

【００６１】

一方、図８に例示されるように、二次多項式補間の方が一次多項式補間よりも処理時間が長い。これは、一次多項式補間では２点のデータを用いて計算を行うのに対し、二次多項式補間では３点のデータを用いて計算を行うので、二次多項式補間の方が一次多項式補間よりも補間の際の計算量が多くなるためである。

【００６２】

また、図８に例示されるように、フィルタの長さＮの値を大きくすると、処理時間が長くなっている。これは、フィルタの長さＮの値を大きくすると、より細かく補間処理がなされて中間音声データのサンプル数が多くなり、フィルタ処理にかかる時間が長くなるためである。

【００６３】

第１の例として、量子化ビット数が大きい場合、例えば量子化ビット数が１６ビット、サンプリング周波数が１２ｋＨｚの出力音声データに対して要求されるＳＮ比が例えば８０ｄＢ以上である場合について考える。この場合、図９から、中間音声データのサンプリ

ング周波数ｆｐが７２ｋＨｚ以上でＳＮ比が８０ｄＢを超えることが分かる。この条件から上述の式（１）における定数ｃを求めると、次のようになる。すなわち、出力音声データのサンプリング周波数が１２ｋＨｚ、入力音声データの量子化ビット数が１６ビットであるこの例では、ｙ＝７２ｋＨｚ／１２ｋＨｚ＝６となり、定数ｃ＝６／１６（ビット）＝０．３７５となる。

【００６４】

定数ｃは、予め求めてＲＯＭ１１２に記憶させたり、本第１の実施形態によるサンプリング周波数をＣＰＵ１１０に実行させるためのプログラムデータに埋め込んでおく。

【００６５】

ここで、値ｙは、出力音声データのサンプリング周波数ｆｏに対する中間音声データのサンプリング周波数ｆｐの倍率である。したがって、フィルタの長さＮ＝ｙ＝６として中間音声データを間引きすると、出力音声データに指定されたサンプリング周波数ｆｏを得ることができ、そのとき、出力音声データに要求される８０ｄＢ以上のＳＮ比を達成することができる。また、量子化ビット数が１６ビットの入力音声データのサンプリング周波数を１１．０２５ｋＨｚから１２ｋＨｚに変換する場合の、８０ｄＢ以上のＳＮ比が得られる最小の処理時間は、図１０から求めることができる。

【００６６】

第２の例として、量子化ビット数が小さい場合について説明する。一例として、量子化ビット数が１２ビット、サンプリング周波数ｆｉが１１．０２５ｋＨｚの入力音声データのサンプリング周波数を１２ｋＨｚに変換する場合について考える。なお、中間音声データを生成する際の補間方法は、上述と同様にラグランジュの一次多項式補間と二次多項式補間とをそれぞれ用い、入力音声データとして、量子化ビット数が１２ビット、再生時間長が３０秒のＰＣＭ形式の音声データを使用している。

【００６７】

量子化ビット数が１２ビットの入力音声データのサンプリング周波数ｆｉを１１．０２５ｋＨｚから１２ｋＨｚに変換する処理における、出力音声データのＳＮ比の実測値の例を図１０に示す。なお、図１０において、縦軸は出力音声データのＳＮ比を示し、横軸は補間処理によって生成された中間音声データのサンプリング周波数ｆｐ（単位：ｋＨｚ）を示す。また、図１０において、破線がラグランジュの一次多項式補間の例、実線が二次多項式補間の例である。

【００６８】

図１０から分かるように、中間音声データを生成する際の補間処理に二次多項式補間を用いた方が、一次多項式補間を用いた場合よりもＳＮ比が高い。ここで、図１０において、二次多項式補間を用いた場合に、中間音声データのサンプリング周波数ｆｐが６０ｋＨｚ以上でＳＮ比の増加が見込めなくなる。これは、サンプリング周波数ｆｐが６０ｋＨｚ以上で、量子化ビット数が１２ビットの場合のＳＮ比の理論的な限界に近づくためである。

【００６９】

また、上述の式（１）によれば、入力音声データの量子化ビット数が小さいときは、中間音声データのサンプリング周波数も低くなるため、中間音声データを生成するための入力音声データに対する補間サンプル数が減ることになる。したがって、入力音声データの量子化ビット数が小さくなるほど、補間処理に要する時間が短縮される。

【００７０】

ここで、上述の式（１）によれば、入力音声データの量子化ビット数が１２ビットの場合には、値ｙ＝４．５となり、中間音声データのサンプリング周波数ｆｐが５４ｋＨｚとなる。これは、図１０から分かるように、量子化ビット数が１２ビットの入力音声データのサンプリング周波数ｆｉを１１．０２５ｋＨｚから１２ｋＨｚに変換した場合の最良のＳＮ比ではない。しかしながら、量子化ビット数が小さい場合には、出力音声データに対して高品質が要求されていないと考えられるため、フィルタ係数の個数（フィルタの長さＮ＝ｙ）を減らして処理のＣＰＵに対する負荷の軽減を図ることができる。例えば、低域

通過フィルタにＦＩＲフィルタを用い、ＣＰＵの処理をタップ毎に時分割で割り当てる場合、フィルタ係数の個数を減らすことでタップ数が減り、ＣＰＵに対する負荷が低減される。

【００７１】

なお、値ｙに小数点以下の端数が発生した場合は、小数点以下の切り上げや切り捨てを行うことが考えられる。

【００７２】

なお、上述では、中間音声データのサンプリング周波数ｆｐを、入力音声データの量子化ビット数の一次関数に基づき求め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、本第１の実施形態では、中間音声データのサンプリング周波数ｆｐが、入力音声データの量子化ビット数が低いほど低くなり、当該量子化ビット数が高いほど高くなるような他の関数を用いることができる。一例として、中間音声データのサンプリング周波数ｆｐと入力音声データの量子化ビット数とをパラメータとして、出力音声データのＳＮ比を計測する。そして、計測結果に基づいた近似などで求めた、中間音声データのサンプリング周波数ｆｐの量子化ビット数に対する関数を用いてもよい。

【００７３】

以上説明したように、本第１の実施形態によれば、入力音声データの量子化ビット数が大きいときには、中間音声データのサンプリング周波数を変換した後で高品質の出力音声データを得ることが可能である。また、入力音声データの量子化ビット数が低いときには、中間音声データのサンプリング周波数も低くなるため、中間音声データを生成するための入力音声データに対する補間サンプル数が減り、ＣＰＵに対する負荷を低減させることが可能である。

【００７４】

＜第２の実施形態＞

次に、本発明の第２の実施形態について説明する。上述した第１の実施形態では、中間音声データのサンプリング周波数ｆｐを、入力音声データの量子化ビット数の関数として求めた。本第２の実施形態では、中間音声データのサンプリング周波数ｆｐを、入力音声データの量子化ビット数と、出力音声データのサンプリング周波数ｆｏに対する中間音声データのサンプリング周波数ｆｐの倍率との関係を示すテーブルを参照して求める。

【００７５】

なお、本第２の実施形態に適用可能なハードウェア、出力音声データのサンプリング周波数を入力するための入力画面、音声データのデータ構造は、図１～図３を用いて説明した第１の実施形態と同様のものを適用できるので、ここでの説明を省略する。

【００７６】

表１は、入力音声データの量子化ビット数と、出力音声データのサンプリング周波数に対する中間音声データのサンプリング周波数ｆｐの倍率との関係を示すテーブルの一例を示す。この例では、テーブルにおいて、出力音声データのサンプリング周波数に対する中間音声データのサンプリング周波数ｆｐの倍率（以下、適宜「倍率」と呼ぶ）を、入力音声データの量子化ビット数の範囲に対して関連付ける。

【００７７】

（表１）

ビット数	出力サンプリング周波数に対する倍率
1～8	4
9～16	6
17～24	8

【００７８】
例えば、このテーブルにおいて、設定された量子化ビット数の範囲のうち最も大きな量子化ビット数の出力音声データが所定以上の品質（ＳＮ比）を得られる最小の倍率を算出し、当該量子化ビット数の範囲に対して関連付けることが考えられる。
【００７９】
表１の例では、量子化ビット数が１ビット～８ビットの範囲の入力音声データに対して倍率「４」が関連付けられ、量子化ビット数が９ビット～１６ビットの範囲の入力音声データに対して倍率「６」が関連付けられている。また、量子化ビット数が１７ビット～２４ビットの範囲の入力音声データに対して倍率「８」が関連付けられている。勿論、入力音声データの量子化ビット数の範囲や、当該範囲に対して関連付けられる倍率は、この例に限定されるものではない。
【００８０】
このテーブルは、予め作成し、ＲＯＭ１１２に記憶させたり、ＣＰＵ１１０が本第２の実施形態によるサンプリング周波数の変換処理を行うためのプログラムに埋め込んでおく。これに限らず、ＣＰＵ１１０によってプログラムに従い表示される所定のユーザインターフェイスを用いて、ユーザ操作によりテーブルを直接的に設定することも可能である。
【００８１】
図１１は、本第２の実施形態によるサンプリング周波数変換の一例の処理を示すフローチャートである。なお、図１１のフローチャートにおいて、上述した図４のフローチャートと対応する処理には同一の符号を付し、詳細な説明を省略する。
【００８２】
図１１に示されるように、ステップＳ３０１で入力音声データの量子化ビット数を取得し、ステップＳ３０２で出力音声データのサンプリング周波数ｆｏを取得したら、処理が本第２の実施形態の特徴であるステップＳ１２０１に移行される。ステップＳ１２０１では、ステップＳ３０１で取得された入力音声データの量子化ビット数に基づき表１に例示したテーブルが参照され、当該量子化ビット数に関連付けられた倍率が取得される。そして、取得した倍率を、ステップＳ３０２で取得した出力音声データのサンプリング周波数ｆｏに乗じて、中間音声データのサンプリング周波数ｆｐを得る。
【００８３】
例えば、量子化ビット数が１６ビット、サンプリング周波数ｆｉが１１．０２５ｋＨｚの入力音声データのサンプリング周波数を、１２ｋＨｚに変換する場合ついて考える。この場合、入力音声データの量子化ビット数である１６ビットに基づきテーブルを参照し、対応する倍率「６」を得る。そして、出力音声データのサンプリング周波数である１２ｋＨｚに対してこの倍率「６」を乗じて、中間音声データのサンプリング周波数ｆｐ＝７２ｋＨｚを得る。
【００８４】
ステップＳ１２０１で中間音声データのサンプリング周波数ｆｐが得られた後の処理は、上述した図４におけるステップＳ３０４移行の処理と何ら変わるところがないため、説明を省略する。
【００８５】
以上説明したように、本第２の実施形態によれば、入力音声データの量子化ビット数が大きいときには、中間音声データのサンプリング周波数を変換した後で高品質の出力音声データを得ることが可能である。また、入力音声データの量子化ビット数が低いときには、中間音声データのサンプリング周波数も低くなるため、中間音声データを生成するための入力音声データに対する補間サンプル数が減り、ＣＰＵに対する負荷を低減させることが可能である。
【００８６】
また、本第２の実施形態によれば、入力音声データの量子化ビット数と、中間音声データのサンプリング周波数ｆｐの出力音声データのサンプリング周波数ｆｏに対する倍率との対応関係を、ユーザが設定することができる。したがって、中間音声データのサンプリ

ング周波数ｆｐをユーザが設定することができる。これは、出力音声データの品質を、ユーザが指定することができることを意味する。

【００８７】

＜他の実施形態＞

上述の各実施形態は、システム或は装置のコンピュータ（或いはＣＰＵ、ＭＰＵなど）によりソフトウェア的に実現することも可能である。従って、上述の実施形態をコンピュータで実現するために、該コンピュータに供給されるコンピュータプログラム自体も本発明を実現するものである。つまり、上述の実施形態の機能を実現するためのコンピュータプログラム自体も本発明の一つである。

【００８８】

なお、上述の各実施形態を実現するためのコンピュータプログラムは、コンピュータで読み取り可能であれば、どのような形態であってもよい。例えば、オブジェクトコード、インタプリタにより実行されるプログラム、ＯＳに供給するスクリプトデータ等で構成することができるが、これらに限るものではない。

【００８９】

上述の各実施形態を実現するためのコンピュータプログラムは、記憶媒体又は有線／無線通信によりコンピュータに供給される。プログラムを供給するための記憶媒体としては、例えば、フレキシブルディスク、ハードディスク、磁気テープ等の磁気記憶媒体、ＭＯ、ＣＤ、ＤＶＤ等の光／光磁気記憶媒体、不揮発性の半導体メモリなどがある。

【００９０】

有線／無線通信を用いたコンピュータプログラムの供給方法としては、コンピュータネットワーク上のサーバを利用する方法がある。この場合、本発明を形成するコンピュータプログラムとなりうるデータファイル（プログラムファイル）をサーバに記憶しておく。プログラムファイルとしては、実行形式のものであっても、ソースコードであっても良い。

【００９１】

そして、このサーバにアクセスしたクライアントコンピュータに、プログラムファイルをダウンロードすることによって供給する。この場合、プログラムファイルを複数のセグメントファイルに分割し、セグメントファイルを異なるサーバに分散して配置することも可能である。

【００９２】

つまり、上述の各実施形態を実現するためのプログラムファイルをクライアントコンピュータに提供するサーバ装置も本発明の一つである。

【００９３】

また、上述の各実施形態を実現するためのコンピュータプログラムを暗号化して格納した記憶媒体を配布し、所定の条件を満たしたユーザに、暗号化を解く鍵情報を供給し、ユーザの有するコンピュータへのインストールを許可してもよい。鍵情報は、例えばインターネットを介してホームページからダウンロードさせることによって供給することができる。

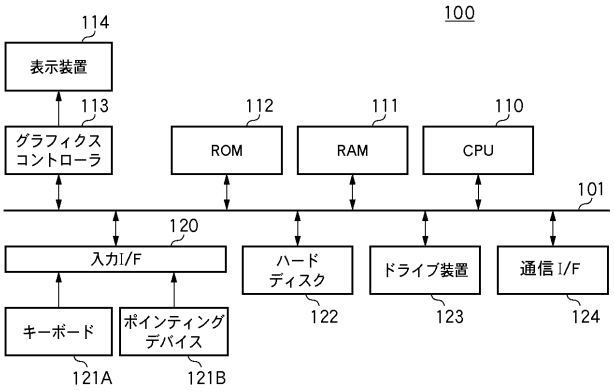
【００９４】

また、上述の各実施形態を実現するためのコンピュータプログラムは、すでにコンピュータ上で稼働するＯＳの機能を利用するものであってもよい。

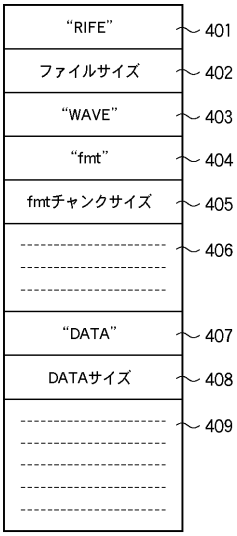
【００９５】

さらに、上述の各実施形態を実現するためのコンピュータプログラムは、その一部をコンピュータに装着される拡張ボード等のファームウェアで構成してもよいし、拡張ボード等が備えるＣＰＵで実行するようにしてもよい。

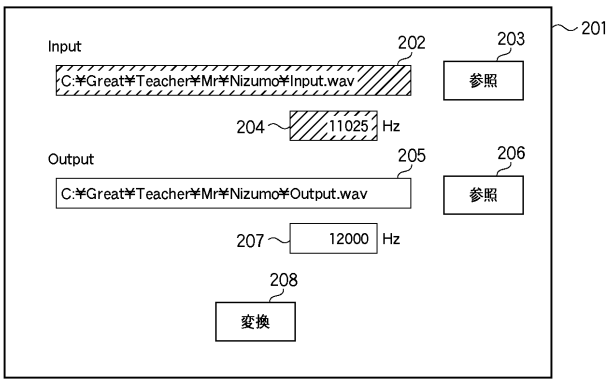
【図１】


100
114
表示装置
113
グラフィクス
コントローラ
112
ROM
111
RAM
110
CPU
101
120
入力I/F
ハード
ディスク
ドライブ装置
通信 I/F
122
123
124
キーボード
ポインティング
デバイス
121A
121B

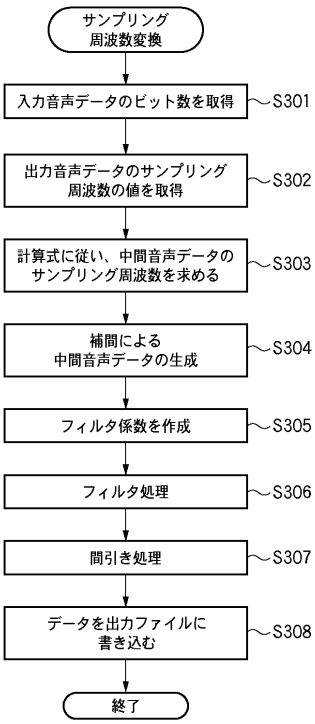
【図２】


“RIFE” 401
ファイルサイズ 402
“WAVE” 403
“fmt” 404
fmtチャンクサイズ 405
406
“DATA” 407
DATAサイズ 408
409

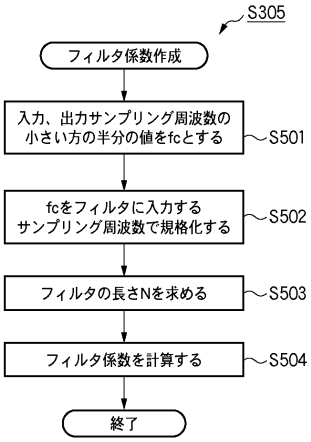
【図３】


201
Input
202
203
C:¥Great¥Teacher¥Mr¥Nizumo¥Input.wav
参照
204
11025 Hz
Output
205
206
C:¥Great¥Teacher¥Mr¥Nizumo¥Output.wav
参照
207
12000 Hz
208
変換

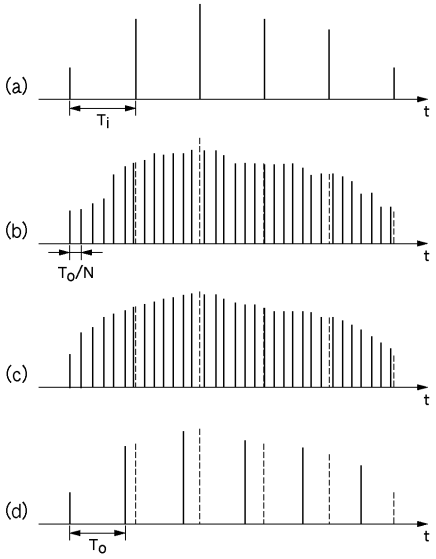
【図４】


サンプリング
周波数変換
入力音声データのビット数を取得 S301
出力音声データのサンプリング
周波数の値を取得 S302
計算式に従い、中間音声データの
サンプリング周波数を求める S303
補間による
中間音声データの生成 S304
フィルタ係数を作成 S305
フィルタ処理 S306
間引き処理 S307
データを出力ファイルに
書き込む S308
終了

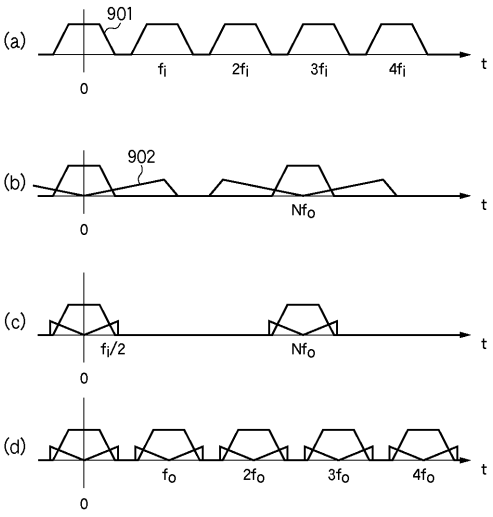
【図５】



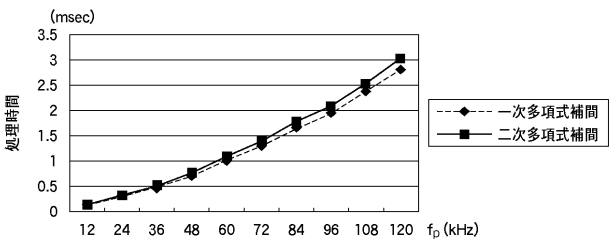
【図６】



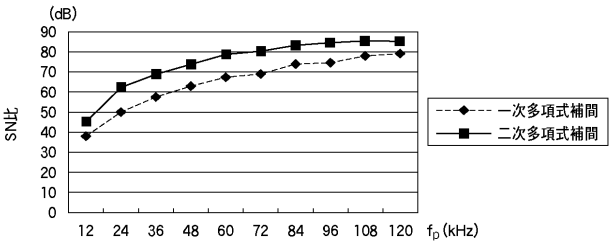
【図７】



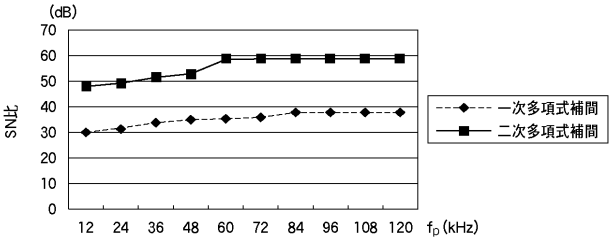
【図８】



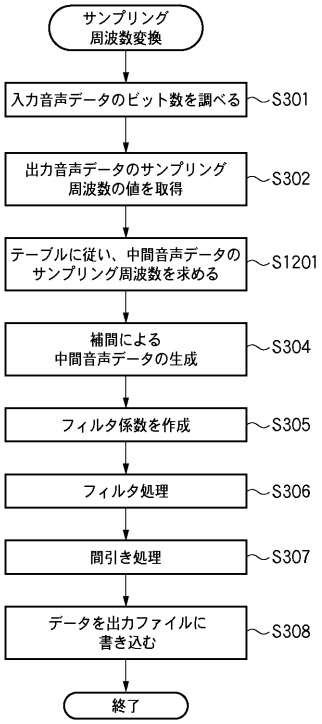
【図９】



【図１０】



【図１１】


サンプリング
周波数変換
入力音声データのビット数を調べる
S301
出力音声データのサンプリング
周波数の値を取得
S302
テーブルに従い、中間音声データの
サンプリング周波数を求める
S1201
補間による
中間音声データの生成
S304
フィルタ係数を作成
S305
フィルタ処理
S306
間引き処理
S307
データを出力ファイルに
書き込む
S308
終了

フロントページの続き

(72)発明者　水野　高志
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