(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6476647号
(P6476647)
(45)発行日　平成31年3月6日(2019.3.6)　(24)登録日　平成31年2月15日(2019.2.15)

(51)Int.Cl.　F I
H03M 7/40 (2006.01)　H03M 7/40

請求項の数 9　(全 29 頁)

(21)出願番号　特願2014-167895(P2014-167895)
(22)出願日　平成26年8月20日(2014.8.20)
(65)公開番号　特開2016-46602(P2016-46602A)
(43)公開日　平成28年4月4日(2016.4.4)
審査請求日　平成29年5月11日(2017.5.11)

前置審査

(73)特許権者　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(74)代理人　110002147
特許業務法人酒井国際特許事務所
(72)発明者　片岡　正弘
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(72)発明者　東　秀人
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(72)発明者　小澤　崇記
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
最終頁に続く

(54)【発明の名称】圧縮プログラム、圧縮装置、圧縮方法、伸長プログラム、伸長装置および伸長方法

(57)【特許請求の範囲】
【請求項１】
コンピュータに、
数値を示す、少なくとも１つの数字を含む数値文字列と、該数値文字列以外の単語とを含む入力データから、前記数値文字列を抽出し、
数値文字列と圧縮符号とを対応付けて記憶する辞書を参照し、前記抽出された数値文字列に対応する圧縮符号に変換し、
前記辞書は、所定の複数の数値文字列を予め記憶する第１の辞書と、前記第１の辞書に登録されている数値文字列を記憶する第２の辞書とであり、
前記抽出された数値文字列が、前記第１の辞書に登録されており、かつ、該数値文字列に対応する圧縮符号が前記第１の辞書に登録されていない場合、該数値文字列に圧縮符号を付与して前記第１の辞書における該数値文字列に対応付けて該圧縮符号を登録し、
前記変換する処理は、前記登録された圧縮符号に対応する数値文字列が抽出された場合に、前記抽出された数値文字列を、前記登録された圧縮符号に変換する
処理を実行させることを特徴とする圧縮プログラム。
【請求項２】
前記第１の辞書は、抽出され得る数値文字列とともに、数値の大きさの順番に応じて割り当てられるコードがさらに対応付けて記憶された辞書であることを特徴とする請求項１に記載の圧縮プログラム。
【請求項３】

前記抽出する処理で数値文字列が抽出された際に、前記数値文字列の検索対象となる複数の圧縮ファイルの各々が、前記数値文字列に対応する圧縮符号を格納するか否かを、前記圧縮ファイル毎に情報ビット列により示すインデックスを参照することにより、前記圧縮符号を格納する圧縮ファイルを特定し、前記情報ビット列において該圧縮ファイルに対応する情報ビットが未格納を示す値の場合、格納を示す値に更新する処理をさらにコンピュータに実行させることを特徴とする請求項１または２に記載の圧縮プログラム。

【請求項４】

数値を示す、少なくとも１つの数字を含む数値文字列と、該数値文字列以外の単語とを含む入力データから、前記数値文字列を抽出する抽出部と、

数値文字列と圧縮符号とを対応付けて記憶する辞書を参照し、前記抽出された数値文字列に対応する圧縮符号に変換する変換部とを有し、

前記辞書は、所定の複数の数値文字列を予め記憶する第１の辞書と、前記第１の辞書に登録されている数値文字列を記憶する第２の辞書とであり、

前記抽出された数値文字列が、前記第１の辞書に登録されており、かつ、該数値文字列に対応する圧縮符号が前記第１の辞書に登録されていない場合、該数値文字列に圧縮符号を付与して前記第１の辞書における該数値文字列に対応付けて該圧縮符号を登録し、

前記変換する処理は、前記登録された圧縮符号に対応する数値文字列が抽出された場合に、前記抽出された数値文字列を、前記登録された圧縮符号に変換する

ことを特徴とする圧縮装置。

【請求項５】

コンピュータが、

数値を示す、少なくとも１つの数字を含む数値文字列と、該数値文字列以外の単語とを含む入力データから、前記数値文字列を抽出し、

数値文字列と圧縮符号とを対応付けて記憶する辞書を参照し、前記抽出された数値文字列に対応する圧縮符号に変換し、

前記辞書は、所定の複数の数値文字列を予め記憶する第１の辞書と、前記第１の辞書に登録されている数値文字列を記憶する第２の辞書とであり、

前記抽出された数値文字列が、前記第１の辞書に登録されており、かつ、該数値文字列に対応する圧縮符号が前記第１の辞書に登録されていない場合、該数値文字列に圧縮符号を付与して前記第１の辞書における該数値文字列に対応付けて該圧縮符号を登録し、

前記変換する処理は、前記登録された圧縮符号に対応する数値文字列が抽出された場合に、前記抽出された数値文字列を、前記登録された圧縮符号に変換する

処理を実行することを特徴とする圧縮方法。

【請求項６】

コンピュータに、

数値を示す、少なくとも１つの数字を含む数値文字列とともに数値の大きさの順番に応じて割り当てられるコードと圧縮符号とを対応付けて記憶する辞書を用いて、圧縮ファイルから抽出した圧縮符号をコードに変換し、

前記変換したコードと、数値文字列の検索範囲に対応する１以上のコードとを比較することで、前記変換したコードが前記検索範囲内に含まれるか否かを判定し、

前記検索範囲内に含まれると判定した場合に、前記変換したコードを数値文字列に伸長して表示する

処理を実行させることを特徴とする伸長プログラム。

【請求項７】

前記数値文字列の検索対象となる複数の圧縮ファイルの各々が、前記数値文字列に対応する圧縮符号を格納するか否かを、前記圧縮ファイル毎に情報ビット列により示すインデックスを参照することにより、前記検索範囲内の数値文字列を含む圧縮ファイルを特定する処理をさらに実行することを特徴とする請求項６に記載の伸長プログラム。

【請求項８】

数値を示す、少なくとも１つの数字を含む数値文字列の大きさの順番に応じて割り当て

られるコードと圧縮符号とを対応付けて記憶する辞書を用いて、圧縮ファイルから抽出した圧縮符号をコードに変換する変換部と、
前記変換したコードと、数値文字列の検索範囲に対応する１以上のコードとを比較することで、前記変換したコードが前記検索範囲内に含まれるか否かを判定する判定部と、
前記検索範囲内に含まれると判定した場合に、前記変換したコードを数値文字列に伸長して表示する伸長部と
を有することを特徴とする伸長装置。
【請求項９】
コンピュータが、
数値を示す、少なくとも１つの数字を含む数値文字列の大きさの順番に応じて割り当てられるコードと圧縮符号とを対応付けて記憶する辞書を用いて、圧縮ファイルから抽出した圧縮符号をコードに変換し、
前記変換したコードと、数値文字列の検索範囲に対応する１以上のコードとを比較することで、前記変換したコードが前記検索範囲内に含まれるか否かを判定し、
前記検索範囲内に含まれると判定した場合に、前記変換したコードを数値文字列に伸長して表示する処理を実行することを特徴とする伸長方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、圧縮プログラムおよび伸長プログラム等に関する。
【背景技術】
【０００２】
圧縮対象のテキストファイルから数値を抽出し、抽出した数値を数字単位で圧縮する技術が存在する。かかる技術では、０～９までの各数字に圧縮符号を割り当て、圧縮対象のテキストファイルから抽出した数値に含まれる各数字を圧縮符号に変換する。さらに、変換された圧縮符号に、例えば全角または半角の指定、カンマの有無、小数点の有無、有効桁数等の数値に関する情報を表す符号を付加することで数値を圧縮符号に変換する。
【先行技術文献】
【特許文献】
【０００３】
【特許文献１】国際公開第２００８／０４７４３２号
【特許文献２】特開２０１３－１５００４１号公報
【特許文献３】特開平０５－１７４０６４号公報
【発明の概要】
【発明が解決しようとする課題】
【０００４】
しかしながら、数値を数字単位で圧縮すると、桁数分の数字に対応した圧縮符号が必要となり、数値全体の圧縮符号が長くなるため、圧縮率が低下する場合がある。
【０００５】
一つの側面では、数値を圧縮する場合の圧縮率を向上させる圧縮プログラムおよび伸長プログラム等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】
【０００６】
第１の案では、圧縮プログラムは、コンピュータに、入力されたデータから、数値を示す、少なくとも１つの数字を含む数値文字列を抽出する。圧縮プログラムは、コンピュータに、数値文字列と圧縮符号とを対応付けて記憶する辞書を参照し、抽出された数値文字列に対応する圧縮符号に変換する処理を実行させる。
【発明の効果】
【０００７】
本発明の１実施態様によれば、圧縮処理時に数値に割り当てる符号長を短くできるとい

う効果を奏する。
【図面の簡単な説明】
【０００８】
【図１】図１は、実施例１の数値の圧縮処理を説明するための図である。
【図２】図２は、実施例１の圧縮処理の全体の流れについて説明するための図である。
【図３Ａ】図３Ａは、参考例１の符号長を説明するための図である。
【図３Ｂ】図３Ｂは、実施例１の符号長を説明するための図である。
【図４】図４は、参考例１および実施例１の符号長の比較を説明するための図である。
【図５】図５は、情報処理装置のシステム構成の例を説明するための図である。
【図６】図６は、実施例１の圧縮処理に係るシステム構成の例を示す図である。
【図７】図７は、ビットフィルタの数値部の第１の例を示す図である。
【図８】図８は、数値文字列に割り当てられる単語コードのデータ構造の例を示す図である。
【図９】図９は、ビットフィルタの単語部の例を示す図である。
【図１０】図１０は、動的辞書の一例を示す図である。
【図１１】図１１は、ビットフィルタの数値部の第２の例を示す図である。
【図１２】図１２は、実施例１の圧縮処理の流れの例を示す図である。
【図１３】図１３は、実施例２の圧縮処理に係るシステム構成の例を示す図である。
【図１４】図１４は、ビットマップ型全文インデックスの一例を示す図である。
【図１５】図１５は、実施例２の圧縮処理の流れを示す図である。
【図１６】図１６は、実施例３の大小比較処理に係るシステム構成の一例を示す図である。
【図１７】図１７は、圧縮ファイル選択の処理の流れの例を示す第１の図である。
【図１８】図１８は、圧縮ファイル選択の処理の流れの例を示す第２の図である。
【図１９】図１９は、伸長辞書の構造を説明するための図である。
【図２０】図２０は、数値文字列の大小比較を説明するための図である。
【図２１】図２１は、実施例３の大小比較処理の全体の流れを示す図である。
【図２２】図２２は、実施例３の圧縮ファイル選択処理の流れを示す図である。
【図２３】図２３は、実施例３の数値文字列の単語コード抽出処理の流れを示す図である。
【図２４】図２４は、実施例１～３の情報処理装置のハードウェア構成を示す図である。
【図２５】図２５は、コンピュータで動作するプログラムの構成例を示す図である。
【図２６】図２６は、実施形態のシステムにおける装置の構成例を示す図である。
【発明を実施するための形態】
【０００９】
以下に、本願の開示する圧縮プログラムの実施例を図面に基づいて詳細に説明する。なお、この実施例によりこの権利範囲が限定されるものではない。各実施例は、処理内容を矛盾させない範囲で適宜組み合わせることが可能である。
【実施例１】
【００１０】
（実施例１の圧縮処理）
図１を用いて、実施例１の情報処理装置１００による圧縮処理について説明する。図１は、実施例１の数値の圧縮処理を説明するための図である。図１の例のように、情報処理装置１００は、圧縮処理の対象である対象ファイル１０に含まれる「He pays 1,200 yen ・・・」を「He」「pays」「1,200」「yen」のように数値または単語ごとに区切り、各数値または各単語を取得する。以下、１以上の数字を有する数値を数値文字列と呼ぶ。数値文字列は、数字以外にプラス・マイナス、カンマおよび小数点等の符号を含んでもよい。情報処理装置１００は、取得した単語のうち数値文字列「1,200」を抽出してビットフィルタの数値部１２１ａに出力する。ビットフィルタの数値部１２１ａは、各数値文字列に対して単語コードと、圧縮符号とを対応付ける辞書である。ビットフィルタの数値部

１２１ａは、それぞれの数値文字列に対応する単語コード１１があらかじめ登録されている。例えば、ビットフィルタの数値部１２１ａには、整数「0」「1」「2」「3」「4」…に対応する単語コード「B00000h」「B00010h」「B00020h」「B00030h」「B00040h」…が順番にあらかじめ登録されている。ビットフィルタの数値部１２１ａのデータ構造に関する詳細は後述する。

【００１１】

なお、数値文字列以外の「He」「pays」「yen」等の単語は、後述するようにビットフィルタの単語部１２１ｂに出力される。また、以降、ビットフィルタの数値部１２１ａをビットフィルタ１２１ａと表記する場合があり、ビットフィルタの単語部１２１ｂをビットフィルタ１２１ｂと表記する。ビットフィルタ１２１ａおよびビットフィルタ１２１ｂの詳細は後述する。

【００１２】

次いで、情報処理装置１００は、ビットフィルタ１２１ａから数値文字列「1,200」に対応する単語コード「B04B01h」を取得する。次いで、情報処理装置１００は、取得した単語コード「B04B01h」と、動的辞書１２２に登録した順に動的に付される圧縮符号「A005h」とを対応づけて動的辞書１２２に登録する。なお、圧縮符号単語コード「B04B01h」および「A005h」等の末尾に示す「h」は、１６進数で表記されていることを示す符号である。

【００１３】

次いで、情報処理装置１００は、動的辞書１２２において動的に付された圧縮符号「A005h」を、数値文字列「1,200」および単語コード「B04B01h」に対応づけてビットフィルタ１２１ａに登録する。そして、情報処理装置１００は、ビットフィルタ１２１ａを基にして数値文字列「1,200」に対応する圧縮符号「A005h」を取得し、圧縮ファイル１２に出力する。

【００１４】

また、情報処理装置１００は、次回、対象ファイル１０に数値文字列「1,200」が出現した場合、既にビットフィルタ１２１ａに登録されている圧縮符号「A005h」を取得し、圧縮ファイル１２に出力する。

【００１５】

図２を用いて実施例１の圧縮処理全体の流れについて説明する。図２は、実施例１の圧縮処理の全体の流れについて説明するための図である。図２の例のように、情報処理装置１００は、対象ファイル１０から単語または数値文字列を抽出する。例えば、情報処理装置１００は、対象ファイル１０から単語「pays」を抽出した場合、単語「pays」を符号化部１Ａに格納する。情報処理装置１００は、ビットフィルタ（単語部）１２１ｂから「pays」の圧縮符号を取得し、記憶領域１Ｂに格納する。

【００１６】

一方、情報処理装置１００は、対象ファイルから数値文字列「1,200」を抽出した場合、数値文字列「1,200」を符号化部２Ａに格納する。情報処理装置１００は、ビットフィルタ（数値部）１２１ａから数値文字列「1,200」の圧縮符号を取得し、記憶領域２Ｂに格納する。情報処理装置１００は、記憶領域１Ｂおよび記憶領域２Ｂに格納した圧縮符号を圧縮ファイル１２に出力する。

【００１７】

このように、情報処理装置１００は、対象ファイル１０から単語を抽出した場合はビットフィルタ１２１ｂを用いて単語を圧縮符号に変換し、数値文字列を抽出した場合はビットフィルタ１２１ａを用いて数値文字列を圧縮符号に変換する。

【００１８】

（参考例１および実施例１の比較）

上述したように実施例１では、数値文字列全体を一つの単位として圧縮する。一方、参考例１では、数値文字列に含まれる個々の数字を一つの単位として圧縮するものとする。参考例１のように数値文字列を数字単位で圧縮すると、ヘッダ、カンマの有無、小数点の

有無等の付加情報を圧縮符号に加えることになる。また、参考例１では、数値文字列が大きくなると数値文字列の桁数に比例して圧縮符号の符号長が長くなる。これに対して、実施例１のように数値文字列単位で圧縮すれば、数値の桁数に関係なく、固定長の圧縮符号が付与され、参考例１と比べて安定して圧縮符号の符号長を短くすることができる。

【００１９】

また、実施例１では、情報処理装置１００は、カンマの有無や小数点の有無等によって、同じ大きさの数値文字列に異なる圧縮符号を付与してもよい。例えば、情報処理装置１００は、カンマが付与されていない「1200」と、カンマが付与された「1,200」とで異なる圧縮符号を付与してもよい。

【００２０】

図３Ａ、図３Ｂおよび図４を用いて、参考例１および実施例１で割り当てられる符号長の比較を説明する。図３Ａは、参考例１の符号長を説明するための図である。図３Ａの例のように、数値文字列１α「1,200」を参考例１のように数字単位で圧縮する場合、圧縮符号１βは情報部と数字部とを有する。情報部は、６ビットのヘッダ情報、１ビットのカンマの有無情報、６ビットの有効桁数情報を有する。一方、数字部は、数値文字列の各桁に圧縮符号が割り当てられ、６ビット×数値文字列の桁数がビット数となる。例えば、数値文字列が４桁の場合、数字部のビット数が２４ビットとなる。したがって、圧縮符号１βの符号長は、数値文字列が４桁の場合、情報部と数値部を合わせると３７ビットとなる。

【００２１】

図３Ｂは、実施例１の符号長を説明するための図である。実施例１に係る情報処理装置１００は、例えば、対象ファイルから数値文字列を抽出し、抽出した数値文字列に対して１６ビットの圧縮符号を割当てる。すなわち、情報処理装置１００は、対象のファイルから抽出した数値文字列に対し、抽出した順番に１６ビットの圧縮符号「A000h」「A001h」「A002h」「A003h」…を割当てる。

【００２２】

実施例１においては、対象のファイルから抽出した数値文字列が圧縮符号の符号長が１６ビットの固定長となる。このため、実施例１に係る情報処理装置１００は、参考例１と比べて安定して短い符号長の圧縮符号を割当てることができる。

【００２３】

例えば、図３Ｂの例のように、数値文字列１α「1,200」を実施例１のように数値単位で圧縮する場合、情報処理装置１００は、数値文字列１α「1,200」に圧縮符号２βとして１６ビットの固定長の圧縮符号「1010000000000101」(A005h)を付与する。

【００２４】

図４は、参考例１および実施例１の符号長の比較を説明するための図である。表１γは、数値文字列の桁数に対応する符号長を示す図である。例えば、圧縮する数値文字列が１桁の場合に、実施例１のように数値文字列単位で圧縮すると符号長が１６ビットとなり、参考例１のように数字単位で圧縮すると、情報部が１３ビットで数字部が６ビットとなり符号長が全体で１９ビットとなる。例えば、圧縮する数値文字列が３桁の場合に、実施例１のように数値文字列単位で圧縮すると符号長が１６ビットとなり、参考例１のように数字単位で圧縮すると、情報部が１３ビットで数字部が１８ビットとなり符号長が全体で３１ビットとなる。また、数値文字列が５桁の場合に、数値文字列単位で圧縮すると符号長が１６ビットとなり、数字文字列単位で圧縮すると符号長が４３ビットとなる。なお、参考例１において数字１桁に割り当てられる符号長は６ビットであるものとする。

【００２５】

このように、参考例１のように数字単位で圧縮する場合、ヘッダ情報等を有する情報部がある分、圧縮符号の符号長が長くなる。また、数値文字列の桁数が大きくなると、各桁に所定長の圧縮符号が割り当てられ、さらに圧縮符号全体の符号長が長くなる。これに対して、実施例１のように数値文字列単位で圧縮する場合、例えば、対象ファイルに出現した順番に昇順に１６ビットの固定長の圧縮符号を割当てるので、数値の桁数に関係なく数

値文字列に割当てる符号長を安定して短く設定することができる。

【００２６】

（実施例１の圧縮処理に関する処理部の構成）

図５を用いて、情報処理装置１００の圧縮部と記憶部との関係について説明する。図５は、情報処理装置のシステム構成の例を説明するための図である。図５の例に示すように、情報処理装置１００の記憶部１２０は、圧縮部１１０と処理部１５０とに接続される。記憶部１２０は例えば、ＲＡＭ（Random Access Memory）、ＲＯＭ（Read Only Memory）、フラッシュメモリなどの半導体メモリ素子、ハードディスクや光ディスクなどの記憶装置に対応する。

【００２７】

また、情報処理装置１００は、圧縮部１１０と、処理部１５０とを有する。圧縮部１１０および処理部１５０の機能は例えば、ＣＰＵ（Central Processing Unit）が所定のプログラムを実行することで実現することができる。また、圧縮部１１０および処理部１５０の機能は例えば、ＡＳＩＣ（Application Specific Integrated Circuit）やＦＰＧＡ（Field Programmable Gate Array）などの集積回路により実現することができる。

【００２８】

圧縮部１１０は、入力されたデータから、数値を示す、少なくとも１つの数字を含む数値文字列を抽出する。圧縮部１１０は、数値文字列と圧縮符号とを対応付けて記憶する辞書を参照し、抽出された数値文字列に対応する圧縮符号に変換する。

【００２９】

辞書は、所定の複数の数値文字列を予め記憶する第１の辞書と、第１の辞書に登録されている数値文字列を記憶する第２の辞書とである。圧縮部１１０は、抽出された数値文字列が、第１の辞書に登録されており、かつ、該数値文字列に対応する圧縮符号が第１の辞書に登録されていない場合、該数値文字列に圧縮符号を付与して第１の辞書における該数値文字列に対応付けて該圧縮符号を登録する。なお、第１の辞書は、例えば、ビットフィルタであり、第２辞書は、例えば、動的辞書である。

【００３０】

第１の辞書は、抽出され得る数値文字列とともに、数値の大きさの順番に応じて割り当てられるコードがさらに対応付けて記憶された辞書である。

【００３１】

図６を用いて、実施例１の圧縮処理に係るシステム構成について説明する。図６は、実施例１の圧縮処理に係るシステム構成の一例を示す図である。図６の例に示されるように、情報処理装置１００は、圧縮部１１０と、記憶部１２０とを有する。圧縮部１１０は、ファイルリード部１１１、圧縮符号付与部１１２およびファイルライト部１１３を有する。記憶部１２０は、ビットフィルタ１２１および動的辞書１２２を有する。ビットフィルタ１２１は、数値部１２１ａおよび単語部１２１ｂを有する。以下、実施例１の圧縮部１１０および記憶部１２０の構成について詳細に説明する。

【００３２】

（記憶部の各構成）

ビットフィルタの数値部１２１ａについて説明する。ビットフィルタ１２１ａは、数値文字列に単語コードおよび圧縮符号を対応付けた辞書である。図７は、ビットフィルタの数値部の第１の例を示す図である。図７の例のように、ビットフィルタ１２１ａは、２グラムと、ビットマップと、数値文字列と、文字列長と、単語コードと、登録番号と、圧縮符号とを対応付ける。「２グラム」は、各単語に含まれる連数字である。例えば「115」は、「11」「15」に対応する２グラムを有する。

【００３３】

「ビットマップ」は、２グラムの文字列に対応するビットマップを表す。例えば、「115」は、数値文字列へのポインタによって、２グラム「11」のビットマップ「０＿１＿０＿０＿０」と、２グラム「15」のビットマップ「０＿０＿１＿０＿０」とに対応付けられ

る。例えば、情報処理装置１００は、対象ファイルから「115」を取得した場合に、２グラム「11」のビットマップ「０＿１＿０＿０＿０」と、２グラム「15」のビットマップ「０＿０＿１＿０＿０」とを用いて数値文字列「115」にアクセスする。

【００３４】

「数値文字列」は、ビットフィルタ１２１ａに登録された数値文字列である。「数値文字列」は、「0」「1」「2」…「100」「101」…「999」「1000」…のように数値文字列が連番で登録される。また、「数値文字列」は、３桁ごとにカンマ「,」を有する数値文字列を含んでもよい。例えば、図４の例のようにビットフィルタ１２１ａは、カンマを有する文字列として「1,000」「1,001」「1,002」…を有する。なお、「数値文字列」は、負の値や小数点以下の値を有してもよい。「文字列長」は、各数値文字列の桁数である。

【００３５】

「単語コード」は、各数値文字列に割り当てるコードである。「単語コード」は、数値の大きさの順に、昇順に割当てられる。例えば、「単語コード」は、昇順に列挙された数値文字列「0」「1」「2」「3」「4」…に対して、単語コード「B00000h」「B00010h」「B00020h」「B00030h」…を昇順にそれぞれ割り当てられる。なお、６桁の１６進数で表される単語コードの末尾１桁は、半角／全角の別、カンマの有無、小数点の有無、正の値／負の値の別などを示す情報ビットである。単語コードの下２桁以上の領域が数値文字列に対応する。このように、各数値文字列に対して昇順に数値文字列の単語コードを割当てることで、各数値文字列に対して、数値文字列の大きさに応じた単語コードが割り当てられる。

【００３６】

また、「単語コード」は、ビットフィルタ１２１ａがカンマ、負の値、小数点以下の値を有する数値文字列を含む場合、それぞれに別々の単語コードが割り当てられる。例えば、図７の例のように、「1000」に対して単語コード「B03E80h」が付され、「1,000」に対して単語コード「B03E81h」が割り当てられる。なお、図７の例では、各項目のデータがレコードとして関連づけられて記憶されている例を示したが、上記説明において互いに関連づけられた項目どうしの関係が保たれれば、データは他の記憶のされ方をしても構わない。後述する図９～１１に示されるビットフィルタおよび動的辞書に関しても同様である。また、「登録番号」「圧縮符号」に関する詳細は後述する。

【００３７】

各数値文字列に割り当てられる単語コードの例に関して説明する。図８は、数値文字列に割り当てられる単語コードのデータ構造の例を示す図である。ビットフィルタ１２１ａにおいて、あらかじめ各数値文字列に対して、３バイト、４バイトまたは５バイトの単語コードが割り当てられる。図８の例には、３バイト、４バイトおよび５バイトの単語コードのコード体系が示される。

【００３８】

図８の例のように、単語コードｃ１は、３バイトコードで「B0000h～B3FFFh」までのコード領域を有する。単語コードｃ１は、ｃ１－１領域、ｃ１－２領域およびｃ１－３領域を有する。ｃ１－１領域は、３バイトコードであることを示す「101100」の固定ビットを有する。ｃ１－２領域は、０から１６，３８３までの整数に対応するビットを有する。ｃ１－３の「＊＊＊＊」は、半角／全角の別、カンマの有無、小数点の有無、正の値／負の値の別などを示す情報ビットである。例えば、ｃ１－３の最後の１ビットが「0」の場合、数値文字列にカンマが含まれないことを示し、ｃ１－３の最後の１ビットが「1」の場合、数値文字列にカンマが含まれることを示す。ｃ２－３およびｃ３－３においても同様である。

【００３９】

単語コードｃ２は、４バイトコードで「B400000h～B7FFFFFh」までのコード領域を有する。単語コードｃ２は、ｃ２－１領域、ｃ２－２領域およびｃ２－３領域を有する。ｃ２－１領域は、４バイトコードであることを示す「101101」の固定ビットを有する。ｃ２－２領域は、１６，３８４から１，０４８，５７５までの整数に対応するビットを有する。

ｃ２－３の「＊＊＊＊」は、ｃ１－３と同様に情報ビットである。単語コードｃ３は、５バイトコードで「B800000h～BBFFFFFh」までのコード領域を有する。ｃ３－１領域は、５バイトコードであることを示す「101001」の固定ビットを有する。ｃ３－２領域は、１，０４８，５７６から１，０７３，７４１，８２３までの整数に対応するビットを有する。ｃ３－３の「＊＊＊＊」は、ｃ１－３と同様に情報ビットである。なお、３バイトコード、４バイトコード、５バイトコードは、整数以外に小数点を有する数値、負の数値、カンマを有する数値等を含んでもよい。

【００４０】

次に、ビットフィルタ１２１の単語部１２１ｂについて説明する。ビットフィルタ１２１の単語部１２１ｂは、基礎単語に単語コードおよび圧縮符号を対応付けた辞書である。図９は、ビットフィルタの単語部の例を示す図である。図９の例のように、ビットフィルタ１２１ｂは、２グラムと、ビットマップと、基礎単語と、文字列長と、出現頻度と、圧縮符号と、単語コードと、登録番号とを対応付ける。「２グラム」「ビットマップ」に関しては、ビットフィルタ１２１ａと同じであるので説明を省略する。「基礎単語」は、ビットフィルタ１２１ｂに登録された単語である。例えば、「基礎単語」とは、辞典、テキスト群等からあらかじめ抽出された約１９万語の単語である。なお、「基礎単語」として登録される単語数は、任意の語数でよい。

【００４１】

「文字列長」は、基礎単語の文字列の長さである。「出現頻度」は、頻度集計用のテキストファイル群において各基礎単語が出現した回数である。ここで、頻度集計用のテキストファイル群とは、対象ファイルとは別に用意された各基礎単語の出現頻度を集計するための１以上のテキストファイルである。

【００４２】

基礎単語の出現頻度の集計について説明する。情報処理装置１００は、頻度集計用のテキストファイルを読込み、頻度集計用のテキストファイル群に存在する単語を適宜抽出してビットフィルタ１２１ｂに登録する。さらに、情報処理装置１００は、ビットフィルタ１２１ｂに登録された各々の基礎単語に関し、頻度集計用のテキストファイル群において出現した回数を出現頻度としてカウントする。例えば、図９の例において、ビットフィルタ１２１ｂは、基礎単語「able」が、頻度集計用のテキストファイル群において「785」回出現したことを示す。

【００４３】

「圧縮符号」は、各基礎単語に割り当てられた圧縮符号である。情報処理装置１００は、出現頻度が高い基礎単語に対してより短い符号長を有する圧縮符号を割り当てる。「単語コード」は、各単語に割り当てるコードである。「登録番号」は、後述する動的辞書１２２に圧縮符号が登録された際に、圧縮符号に一意に付される番号である。「登録番号」は、例えば、動的辞書１２２に登録された順番を示す。

【００４４】

次に、動的辞書１２２について説明する。動的辞書１２２は、単語コードと、単語コードに付与された圧縮符号とを対応付ける辞書である。図１０は、動的辞書の一例を示す図である。図１０の例のように、動的辞書１２２は、圧縮符号とコードとを対応付ける。「コード」は、ビットフィルタ１２１ａまたはビットフィルタ１２１ｂで取得された単語コードである。「圧縮符号」は、例えば動的辞書１２２に単語コードが登録された順番に単語コードに昇順に付与される固定長の圧縮符号である。

【００４５】

例えば、情報処理装置１００が、対象ファイルから「order」「box」「1000」「him」…の順番に単語および数値文字列を抽出した場合、各々の単語および数値文字列の単語コードに対して、圧縮符号「A000h」「A001h」「A002h」「A003h」…を割当てる。そして、情報処理装置１００は、各々の単語および数値文字列に対して割り当てた圧縮符号を動的辞書１２２に登録する。例えば、情報処理装置１００は、オフセット「0x0000h」の位置に圧縮符号「A000h」と「order」の単語コードとを対応付ける。また、情報処理装置１０

０は、オフセット「0x0001h」の位置に圧縮符号「A001h」と「box」の単語コードとを対応付ける。また、情報処理装置１００は、オフセット「0x0002h」の位置に圧縮符号「A002h」と「1000」の単語コードとを対応付ける。なお、図１０において「ｃ（単語）」の表記は、カッコ内の数値文字列または単語に対応する単語コードを表す。例えば、「ｃ（order）」は、「order」の単語コードを表す。

【００４６】

（圧縮部の各構成）

圧縮部１１０の各構成について説明する。ファイルリード部１１１は、対象ファイルを読みだして、対象ファイルから単語および数値文字列を抽出する処理部である。ファイルリード部１１１は、対象ファイルの文字列中の空白文字によって文字列を単語または数値文字列ごとに区切り、対象ファイルから各単語および各数値文字列を抽出する。ファイルリード部１１１は、抽出した単語および数値文字列を圧縮符号付与部１１２に出力する。

【００４７】

圧縮符号付与部１１２は、対象ファイルから抽出された単語および数値文字列に圧縮符号を付与する処理部である。圧縮符号付与部１１２は、ファイルリード部１１１から数値文字列を受け付けると、ビットフィルタ１２１ａに登録されている「数値文字列」にアクセスする。

【００４８】

圧縮符号付与部１１２は、アクセスした「数値文字列」に対応する圧縮符号がビットフィルタ１２１ａに既に登録されていた場合、ビットフィルタ１２１ａから数値文字列に対応する圧縮符号を取得してファイルライト部１１３に出力する。

【００４９】

一方、圧縮符号付与部１１２は、アクセスした数値文字列に対応する圧縮符号がビットフィルタ１２１ａに登録されていない場合、ビットフィルタ１２１ａから数値文字列に対応する単語コードを取得する。次いで、圧縮符号付与部１１２は、ビットフィルタ１２１ａから取得した単語コードを、動的辞書１２２への登録順に付与される圧縮符号に対応付けて動的辞書１２２に登録する。

【００５０】

図１０を用いて、動的辞書１２２への圧縮符号の登録の具体例について説明する。図１０の例のように圧縮符号付与部１１２は、３番目に登録された「1000」の単語コードに圧縮符号「A002h」を対応づけて動的辞書１２２に登録する。また、圧縮符号付与部１１２は、６番目に登録された「1,200」の単語コードに圧縮符号「A005h」を対応づけて動的辞書１２２に登録する。このように、圧縮符号付与部１１２は、各数値文字列の単語コードに対して動的辞書１２２へ登録された順番に対応する圧縮符号を付与し、数値文字列の単語コードと、付与された圧縮符号とを対応付けて動的辞書１２２に登録する。

【００５１】

次いで、圧縮符号付与部１１２は、動的辞書１２２に登録した圧縮符号を、単語コードに対応付けてビットフィルタ１２１ａに登録する。さらに、圧縮符号付与部１１２は、登録した圧縮符号に登録番号を付与し、登録番号を圧縮符号に対応付けて動的辞書１２２に登録する。なお、登録番号とは、動的辞書１２２に登録された順番を表す番号である。

【００５２】

図１１を用いて、登録番号および圧縮符号が登録された後のビットフィルタ１２１ａの具体例について説明する。図１１は、ビットフィルタの数値部の第２の例を示す図である。図１１の例のように、ビットフィルタ１２１ａに登録番号および圧縮符号が登録される。ビットフィルタ１２１ａにおいて、「登録番号」は、単語コードが動的辞書１２２に登録された順番を表す番号である。「圧縮符号」は、数値文字列に対応する圧縮符号である。例えば、ビットフィルタ１２１ａにおいて数値文字列「115」に対応する登録番号「15」は、数値文字列「115」に係る圧縮符号「A017」が１５番目に動的辞書１２２に登録されたことを表す。ビットフィルタ１２１ａには、数値文字列「115」に付与された圧縮符号「A017h」が登録される。また、ビットフィルタ１２１ａにおいて数値文字列「121」に

対応する登録番号「12」は、数値文字列「121」に係る圧縮符号「A00E」が１２番目に動的辞書１２２に登録されたことを表す。ビットフィルタ１２１ａには、数値文字列「121」に付与された圧縮符号「A00Eh」が登録される。

【００５３】

そして、圧縮符号付与部１１２は、ビットフィルタ１２１ａに登録した圧縮符号をファイルライト部１１３に出力する。

【００５４】

ファイルライト部１１３は、圧縮符号付与部１２２から出力された圧縮符号を基に圧縮ファイルを生成する処理部である。ファイルライト部１１３は、例えば圧縮符号付与部１２２から出力された数値文字列または単語の各圧縮符号を、それぞれバッファに格納して圧縮データを生成する。ファイルライト部１１３は、バッファに生成された圧縮データを基に圧縮ファイルを生成する。

【００５５】

（実施例１の圧縮処理の流れ）

次に、実施例１の圧縮処理の流れについて説明する。図１２は、実施例１の圧縮処理の流れを説明するための図である。図１２の例のように、情報処理装置１００は、前処理をおこなう（ステップＳ１０）。例えば、情報処理装置１００は、前処理においてビットフィルタ１２１を保持する領域や、動的辞書１２２を作成する作業領域を確保する。ファイルリード部１１１は、対象ファイルを読み出し（ステップＳ１１）、対象ファイルから数値文字列を抽出する（ステップＳ１２）。

【００５６】

圧縮符号付与部１１２は、対象ファイルから抽出された数値文字列に対応する圧縮符号がビットフィルタ１２１ａに登録されているか否かを判定する（ステップＳ１３）。圧縮符号付与部１１２は、ビットフィルタ１２１ａに圧縮符号が登録されている場合（ステップＳ１３Yes）、ステップＳ１８の処理に移行する。

【００５７】

一方、圧縮符号付与部１１２は、ビットフィルタ１２１ａに圧縮符号が登録されていない場合（ステップＳ１３No）、ビットフィルタ１２１ａから単語コードを取得する（ステップＳ１４）。次いで、圧縮符号付与部１１２は、ビットフィルタ１２１ａから取得した単語コードと、動的辞書１２２に単語コードを登録する順に付与される圧縮符号とを対応付けて動的辞書１２２に登録する（ステップＳ１５）。例えば、圧縮符号付与部１１２は、単語コードが動的辞書１２２に登録される順番に、単語コードに圧縮符号「A000h」「A001h」「A002h」「A003h」「A004h」「A005h」・・・を付与する。圧縮符号付与部１１２は、動的辞書１２２から登録された圧縮符号を取得する（ステップＳ１６）。次いで、圧縮符号付与部１１２は、動的辞書１２２から取得された圧縮符号を、単語コードに対応づけてビットフィルタ１２１ａに登録する（ステップＳ１７）。

【００５８】

圧縮符号付与部１１２は、ビットフィルタ１２１ａから数値文字列に対応する圧縮符号を取得する（ステップＳ１８）。ファイルライト部１１３は、ビットフィルタ１２１ａから取得された圧縮符号を圧縮ファイルに書き込む（ステップＳ１９）。

【００５９】

ファイルリード部１１１は、ファイルの読み出し位置がファイルの終端であるか否かを判定する（ステップＳ２０）。ファイルリード部１１１は、読み出し位置がファイルの終端である場合（ステップＳ２０Yes）、処理を終了させる。一方、ファイルリード部１１１は、読み出し位置がファイルの途中である場合（ステップＳ２０No）、ステップＳ１１の処理に戻る。

【００６０】

以上のように実施例１の情報処理装置１００は、対象ファイルに含まれる各数値文字列に圧縮符号を割り当てるので圧縮処理時に数値文字列に割り当てる符号長を短くできる。

【００６１】

（実施例１の効果）
圧縮部１１０は、入力されたデータから、数値を示す、少なくとも１つの数字を含む数値文字列を抽出する。圧縮部１１０は、数値文字列と圧縮符号とを対応付けて記憶する辞書を参照し、抽出された数値文字列に対応する圧縮符号に変換する。これにより、圧縮処理時に数値文字列に割り当てる符号長を短くできる。
【００６２】
辞書は、所定の複数の数値文字列を予め記憶する第１の辞書と、第１の辞書に登録されている数値文字列を記憶する第２の辞書とである。圧縮部１１０は、抽出された数値文字列が、第１の辞書に登録されており、かつ、該数値文字列に対応する圧縮符号が第１の辞書に登録されていない場合、該数値文字列に圧縮符号を付与して第１の辞書における該数値文字列に対応付けて該圧縮符号を登録する。これにより、入力された数値にだけ動的に圧縮符号を付与しつつ圧縮することができ、抽出され得るすべての数値に予め圧縮符号を割当てる場合に比較して、圧縮符号の長さを短くすることができる。
【００６３】
第１の辞書は、抽出され得る数値文字列とともに、数値の大きさの順番に応じて割り当てられるコードがさらに対応付けて記憶された辞書である。これにより、コードの状態で数値の大きさを比較できる。
【実施例２】
【００６４】
図１３を用いて、実施例２の圧縮処理に係るシステム構成について説明する。図１３は、実施例２の圧縮処理に係るシステム構成の例を示す図である。図１３の例に示されるように、情報処理装置２００は、圧縮部２１０と、記憶部２２０とを有する。圧縮部２１０は、ファイルリード部２１１、圧縮符号付与部２１２およびファイルライト部２１３を有する。記憶部２２０は、ビットフィルタ２２１、動的辞書２２２およびビットマップ型全文インデックス２２３を有する。ビットフィルタ２２１は、数値部２２１ａおよび単語部２２１ｂを有する。なお、実施例１と同じ構成に関しては、番号の下２桁を同一にして適宜説明を省略する。
【００６５】
圧縮部２１０は、抽出する処理で数値文字列が抽出された際に、複数の圧縮ファイルのうち数値文字列が含まれる圧縮ファイルを示す情報ビット列を数値文字列ごとに対応付けて保持するインデックスを更新する。以下、実施例２の圧縮部２１０および記憶部２２０の構成について詳細に説明する。
【００６６】
実施例２の情報処理装置２００は、記憶部２２０がビットマップ型全文インデックス２２３を有する点で、実施例１の情報処理装置１００と異なる。図１４を用いてビットマップ型全文インデックス２２３のデータ構造について説明する。図１４は、ビットマップ型全文インデックスの一例を示す図である。図１４の例のように、ビットマップ型全文インデックス２２３は、静的単語および動的単語に係る圧縮付号ごとにビットマップを対応付ける。ビットマップとは、静的単語および動的単語がいずれの圧縮ファイルに含まれるかを表す符号ビット列である。ビットマップの各ビットが、各圧縮ファイルに静的単語または動的単語が含まれているか否かを表す。
【００６７】
ビットマップ型全文インデックス２２３は、例えば、８０００種類の静的単語ごと、および２４０００種類の動的単語ごとにビットマップを対応付ける。静的単語とは、出現頻度集計用のテキストファイル群において各単語の出現頻度を集計した場合に、出現頻度の高い単語を表す。例えば、静的単語は、頻度集計用のテキストファイル群での出現頻度が上位８０００位までの単語である。また、動的単語とは、頻度集計用のテキストファイル群での出現頻度の順位が８０００位未満であって、対象ファイルから抽出された数値文字列または単語である。
【００６８】

例えば、ビットマップ型全文インデックス２２３の有効行１行目は、圧縮符号「0001h」に対応する「ａ」のビットマップが「1011110110…」となっている。ビットマップ型全文インデックスの有効行１行目のビットマップは、「ａ」の圧縮符号が含まれるファイルを表す。ビットマップ「1011110110…」は、１ビット目に「1」が格納されているのでファイル１に「ａ」が含まれ、２ビット目に「0」が格納されているのでファイル２に「ａ」が含まれず、３ビット目に「1」が格納されているのでファイル３に「ａ」が含まれることを表す。また、ビットマップ「1011110110…」は、４ビット目に「1」が格納されているのでファイル４に「ａ」が含まれ、５ビット目に「1」が格納されているのでファイル５に「ａ」が含まれていることを表す。なお、ビットマップ「1011110110…」は、ファイル６以降の他の各ファイルに「ａ」が含まれるか否かについても表す。

【００６９】

次に、ビットマップ型全文インデックス２２３の更新について説明する。ファイルライト部２１３は、圧縮符号付与部２１２から受け付けた圧縮符号がビットマップ型全文インデックス２２３に登録されているか否かを判定する。ファイルライト部２１３は、ビットマップ型全文インデックス２２３に、受け付けた圧縮符号に対応するビットマップが登録されている場合、受け付けた圧縮符号に対応するビットマップを参照する。ファイルライト部２１３は、参照したビットマップのうち、対象ファイルに対応するビットが「0」の場合、ビットを「1」に更新する。なお、ファイルライト部２１３は、対象ファイルに対応するビットが「1」の場合、ビットマップを更新しない。

【００７０】

一方、ファイルライト部２１３は、受け付けた圧縮符号に対応するビットマップがビットマップ型全文インデックス２２３に、登録されていない場合、ビットマップ型全文インデックス２２３に新しくビットマップを登録する。

【００７１】

具体的には、ファイルライト部２１３は、圧縮符号付与部２１２が対象ファイル中の単語または数値文字列を動的辞書２２２に登録した場合に、単語または数値文字列に付与された圧縮符号を取得する。かかる場合において、ファイルライト部２１３は、取得した圧縮符号に係るビットマップをビットマップ型全文インデックス２２３に登録する。このビットマップには、対象ファイルの数分のビット「0」が含まれる。さらに、ファイルライト部２１３は、登録したビットマップのビットのうち、対象ファイルに対応するビットを「1」に更新する。すなわち、ファイルライト部２１３は、対象ファイルにおいて初出の単語または数値文字列を動的辞書２２２に登録した際に、ビットマップ型全文インデックス２２３に登録された単語または数値文字列に対応するビットマップを登録する。このようにしてビットマップ型全文インデックス２２３を生成する。

【００７２】

（実施例２の圧縮処理の流れ）

次に、実施例２の圧縮処理の流れについて説明する。図１５は、実施例２の圧縮処理の流れを説明するための図である。図１５の例のように、情報処理装置２００は、前処理をおこなう（ステップＳ３０）。例えば、情報処理装置２００は、前処理としてビットマップ型全文インデックス２２３を生成するための作業領域を確保する。ファイルリード部２１１は、対象ファイルを読み出し（ステップＳ３１）、対象ファイルから数値文字列を抽出する（ステップＳ３２）。

【００７３】

圧縮符号付与部２１２は、ビットフィルタ２２１ａに、対象ファイルから抽出した数値文字列に対応する圧縮符号が登録されている場合（ステップＳ３３Yes）、ステップＳ３８の処理に移行する。

【００７４】

一方、圧縮符号付与部２１２は、ビットフィルタ２２１ａに、対象ファイルから抽出した数値文字列に対応する圧縮符号が登録されていない場合（ステップＳ３３No）、ビットフィルタ２２１ａから数値文字列の単語コードを取得する（ステップＳ３４）。圧縮符号

付与部２１２は、取得した単語コードと、動的辞書２２２へ登録する順番に付与される圧縮符号とを対応付けて動的辞書２２２に登録する（ステップＳ３５）。圧縮符号付与部２１２は、動的辞書２２２に登録された圧縮符号を取得する（ステップＳ３６）。圧縮符号付与部２１２は、動的辞書２２２から取得した圧縮符号をビットフィルタ２２１ａに登録する（ステップＳ３７）。圧縮符号付与部２１２は、ビットフィルタ２２１ａから圧縮符号を取得する（ステップＳ３８）。

【００７５】

ファイルライト部２１３は、圧縮符号付与部２１２が取得した圧縮符号を基にしてビットマップ型全文インデックス２２３を更新する（ステップＳ３９）。例えば、ファイルライト部２１３は、ビットマップ型全文インデックス２２３に圧縮符号に対応するビットマップが登録されている場合、ビットマップに含まれるビットのうち、対象ファイルに対応するビットを「1」に更新する。一方、ファイルライト部２１３は、ビットマップ型全文インデックス２２３に圧縮符号に対応するビットマップが登録されていない場合、ビットマップ型全文インデックス２２３に新しく圧縮符号に対応するビットマップを登録する。

【００７６】

ファイルライト部２１３は、圧縮符号付与部２１２が取得した圧縮符号を圧縮ファイルに書き込む（ステップＳ４０）。

【００７７】

ファイルリード部２１１は、ファイルの読み出し位置がファイルの終端であるか否かを判定する（ステップＳ４１）。ファイルリード部２１１は、読み出し位置がファイルの終端である場合（ステップＳ４１Yes）、処理を終了させる。一方、ファイルリード部２１１は、読み出し位置がファイルの途中である場合（ステップＳ４１No）、ステップＳ３１の処理に戻る。

【００７８】

以上のように実施例２の情報処理装置２００は、ファイル圧縮を行う際にビットマップ型全文インデックス２２３を生成する。これにより、情報処理装置２００は、複数の圧縮ファイルを基に数値文字列検索をおこなう際に、検索対象の数値文字列を有するファイルを特定でき、オープンする圧縮ファイルを絞り込むことができるので数値文字列検索を高速化することができる。

【００７９】

（実施例２の効果）

圧縮部２１０は、抽出する処理で数値文字列が抽出された際に、複数の圧縮ファイルのうち数値文字列が含まれる圧縮ファイルを示す情報ビット列を数値文字列ごとに対応付けて保持するインデックスを更新する。これにより、複数の圧縮ファイルを基に数値文字列検索をおこなう際に、検索対象の数値文字列を有するファイルを特定でき、オープンする圧縮ファイルを絞り込むことができるので数値文字列検索を高速化することができる。

【実施例３】

【００８０】

数値を数字単位で圧縮された状態で大小比較が可能となるように圧縮すると、出現頻度の高い０、１等と、出現頻度の低い８、９等を同等の長さの符号長で圧縮する必要があり、出現頻度が低い数字に短い符号を割当てることになるから、圧縮ファイル全体として圧縮率が低下する。

【００８１】

例えば、０～９までの各数字に対して４ビットの符号長を割当てる場合、出現頻度が「0.0625」の文字又は単語と同等の符号長が割当てられることになり、出現頻度の低い８、９等の数字が、出現頻度の高い文字または単語と同等に取り扱われる。このため、他の文字または単語に係る圧縮符号の領域が狭められ、他の文字又は単語に割当てる符号長が長く補正されることになるので、圧縮ファイル全体として圧縮率が低下するという問題がある。

【００８２】

図１６を用いて、実施例３の大小比較処理に係るシステム構成について説明する。図１６は、実施例３の大小比較処理に係るシステム構成の一例を示す図である。図１６の例に示されるように、情報処理装置３００は、処理部３３０と、記憶部３２０とを有する。処理部３３０は、検索範囲受付部３３１、ファイル選択部３３２、伸長辞書生成部３３３、ファイルリード部３３４、比較部３３５およびファイルライト部３３６を有する。記憶部３２０は、ビットマップ型全文インデックス３２３および伸長辞書３２４を有する。

【００８３】

記憶部３２０は例えば、ＲＡＭ、ＲＯＭ、フラッシュメモリなどの半導体メモリ素子、ハードディスクや光ディスクなどの記憶装置に対応する。また、処理部３３０の機能は例えば、ＣＰＵが所定のプログラムを実行することで実現することができる。また、処理部３３０の機能は例えば、ＡＳＩＣやＦＰＧＡなどの集積回路により実現することができる。ビットマップ型全文インデックス３２３のデータ構造に関しては、実施例１または実施例２と同じであるので説明を省略する。

【００８４】

処理部３３０は、数値を示す、少なくとも１つの数字を含む数値文字列とともに数値の大きさの順番に応じて割り当てられるコードと圧縮符号とを対応付けて記憶する辞書を用いて、圧縮ファイルから抽出した圧縮符号をコードに変換する。処理部３３０は、変換したコードと、数値文字列の検索範囲に対応する１以上のコードとを比較することで、変換したコードが検索範囲内に含まれるか否かを判定する。処理部３３０は、検索範囲内に含まれると判定した場合に、変換したコードを数値文字列に伸長して表示する。

【００８５】

また、処理部３３０は、各数値文字列を含む圧縮ファイルに関する情報を有するインデックスを用いて、検索範囲内の数値文字列を含む圧縮ファイルを特定する。以下、実施例３の処理部３３０および記憶部３２０の構成について詳細に説明する。

【００８６】

検索範囲受付部３３１は、ユーザによって入力された数値文字列検索の範囲を受け付ける処理部である。検索範囲受付部３３１は、例えば、入力用フォーマットに入力された検索範囲の最大値および最小値を取得することで数値文字列検索の範囲を受け付ける。検索範囲とは、指定された最大値および最小値に属する数値文字列の範囲である。検索範囲受付部３３１は、受け付けた最大値および最小値を検索範囲としてファイル選択部３３２および比較部３３５に出力する。

【００８７】

ファイル選択部３３２は、ビットマップ型全文インデックス３２３を用いて、検索範囲内の数値文字列を有する圧縮ファイルを選択する処理部である。ファイル選択部３３２は、ビットマップ型全文インデックス３２３から、検索範囲内の圧縮符号に対応する１以上のビットマップを抽出する。次いで、ファイル選択部３３２は、抽出した１以上のビットマップ同士でｏｒ演算することで選択結果マップを生成する。選択結果マップとは、検索範囲内の数値文字列を１つ以上含む圧縮ファイルを示すビットマップである。ファイル選択部３３２は、選択結果マップをファイルリード部３３４に出力する。

【００８８】

次に、図１７を用いて選択結果マップを生成する処理の流れを説明する。図１７は、圧縮ファイル選択の処理の流れの例を示す第１の図である。図１７の例には、ビットマップ型全文インデックス３２３の動的単語に対応する部分が示される。図１７の例のように、ファイル選択部３３２は、例えば検索範囲が１１０～１２５であった場合、検索範囲内に属する「115」のビットマップと、「121」のビットマップとを抽出する。次いで、ファイル選択部３３２は、「115」のビットマップ「1011011000…」と、「121」のビットマップ「1001010101…」とでｏｒ演算を行い、選択結果マップ「1011011101…」を生成する。ファイル選択部３３２によって生成された選択結果マップ「1011011101…」は、数値文字列「115」および「121」のいずれか一方または両方を含むそれぞれの圧縮ファイルを示す。

【００８９】

なお、ビットマップ型全文インデックス３２３に、検索範囲内に属するビットマップが３以上のある場合、ファイル選択部３３２は、各ビットマップに対してｏｒ演算を実行して選択結果マップを生成する。例えば、ファイル選択部３３２は、検索範囲内にビットマップＡ、ビットマップＢ、ビットマップＣが属する場合、条件式（Ａ　ｏｒ　Ｂ）　ｏｒ　Ｃを算出することで、選択結果マップを生成する。

【００９０】

なお、ファイル選択部３３２は、ビットマップ型全文インデックス３２３のうち検索範囲内に該当するビットマップを特定する際に、後述する大小比較の方法を用いてもよい。

【００９１】

次に、図１８を用いて圧縮ファイルを選択する処理の流れを説明する。図１８は、圧縮ファイル選択の処理の流れの例を示す第２の図である。図１８の例のように、帳票フォルダ５１には、複数の圧縮ファイル５２ａ、５２ｂ、５２ｃ等が含まれる。ファイル選択部３３２は、選択結果マップ５０が「1011011101…」となっているので、帳票フォルダ５１に含まれるファイルのうち、１番目の圧縮ファイル５２ａと３番目の圧縮ファイル５２ｃとをオープンする。なお、ファイル選択部３３２は、他にも４番目、６～８番目、１０番目のファイルをオープンする。ファイル選択部３３２によってオープンされた圧縮ファイル５２ａには、例えば検索範囲１１０～１２５に属する数値文字列に係る圧縮符号として「115(円)」に係る圧縮符号が格納されている。

【００９２】

伸長辞書生成部３３３は、伸長辞書３２４を生成する処理部である。伸長辞書生成部３３３は、圧縮ファイルに含まれる圧縮データに基づいて伸長辞書３２４を生成する。図１９は、伸長辞書の構造を説明するための図である。図１９の例のように、伸長辞書３２４は根３２４ａと、枝３２４ｂ（１）～３２４ｂ（４）、葉３２４ｃ（１）～３２４ｃ（４）を有する。

【００９３】

枝３２４ｂ（１）～３２４ｂ（４）は、それぞれ３２４ｃ（１）～３２４ｃ（４）に格納された数値文字列に対応する圧縮符号が含まれる。情報処理装置３００は、例えば、圧縮ファイルから圧縮符号を読み込んだ際に、読み込んだ圧縮符号と、枝３２４ｂ（１）～３２４ｂ（４）に含まれる圧縮符号とを比較することで、読み込んだ圧縮符号に対応する葉３２４ｃを特定する。

【００９４】

葉３２４ｃ（１）～３２４ｃ（４）に対応する葉の構造体は、例えば葉の構造体３２４Ｃに表される。葉の構造体３２４Ｃは、葉識別情報、圧縮符号長、文字コードまたは数値テーブルへのポインタ等を有する。「葉識別情報」は、葉を一意に識別する情報である。「圧縮符号長」は、後述するファイルリード部３３４によって取得された圧縮データのビット列のうち、有効な長さを示す情報である。例えば、各数値文字列には、１６ビットの固定長符号が割り当てられるので、数値文字列に対応する圧縮符号長には、「16」ビットが格納される。「文字コード」は、例えばアスキーコード等の文字コードを示す。「数値テーブルへのポインタ」は、複数の数値文字列の単語コードを格納する数値テーブルにおいて、取得された圧縮符号に対応する数値文字列の単語コードが格納されている位置を示すポインタである。なお、葉の構造体３２４Ｃには、文字コードまたは数値テーブルへのポインタのいずれか一方を有する。

【００９５】

数値テーブル３２４ｄについて説明する。数値テーブル３２４ｄは、ビットフィルタ１２１においてそれぞれの数値文字列に割り当てられた全ての単語コードを有する。例えば、数値テーブル３２４ｄは、「110」の単語コード３２４ｄ（１）「B006E0h」、「115」の単語コード３２４ｄ（２）「B00730h」、「125」の単語コード３２４ｄ（３）「B007D0h」を有する。葉３２４ｃ（１）～３２４ｃ（４）はそれぞれ、数値テーブルへのポインタによって数値テーブル３２４ｄに含まれる数値文字列の単語コードに対応付けられる。例えば、数値文字列「110」に対応する葉の構造体３２４Ｃに係る数値テーブルのポイン

タには、領域３２４ｄ（１）の先頭アドレスに対応するオフセットが格納されている。また、数値文字列「115」に対応する葉の構造体３２４Ｃに係る数値テーブルのポインタには、領域３２４ｄ（２）の先頭アドレスに対応するオフセットが格納されている。

【００９６】

ファイルリード部３３４は、圧縮ファイルを読み込む処理部である。ファイルリード部３３４は、圧縮ファイルから圧縮符号を取得する。ファイルリード部３３４は、圧縮ファイルから取得した圧縮符号を伸長辞書３２４と照らし合わせる。すなわち、ファイルリード部３３４は、圧縮ファイルから取得した圧縮符号と、葉３２４ｃ（１）～３２４ｃ（４）とを比較し、圧縮ファイルから取得した圧縮符号に対応する葉の構造体３２４Ｃを特定する。ファイルリード部３３４は、圧縮符号に対応する葉の構造体３２４Ｃにアクセスする。ファイルリード部３３４は、アクセスした葉の構造体３２４Ｃに数値テーブルへのポインタが含まれる場合、数値テーブルへのポインタに基づいて数値テーブル３２４ｄから単語コードを取得する。

【００９７】

例えば、ファイルリード部３３４は、取得した圧縮符号「1010000000010111」が枝３２４ｂ（１）にヒットする場合、葉３２４ｃ（１）にアクセスする。ファイルリード部３３４は、葉３２４ｃ（１）の葉の構造体３２４Ｃに数値テーブルへのポインタが格納されていた場合、数値テーブルへのポインタを取得する。ファイルリード部３３４は、数値テーブルへのポインタに基づいて、数値テーブル３２４ｄ内の領域３２４ｄ（２）の物理アドレスを特定する。ファイルリード部３３４は、領域３２４ｄ（２）から数値文字列「115」に対応する単語コード「B00730h」を取得する。そして、ファイルリード部３３４は、取得した数値文字列「115」に対応する単語コード「B00730h」を比較部３３５に出力する。

【００９８】

比較部３３５は、検索範囲に対応する単語コードと、ファイルリード部３３４によって取得された各単語コードとを比較する処理部である。比較部３３５は、検索範囲の最大値と最小値とに対応する単語コードを取得する。比較部３３５は、伸長辞書３２４から検索範囲の最大値と最小値とに対応する単語コードを取得してもよい。次いで、ファイルリード部３３４は、圧縮ファイルを読込み、適宜、圧縮ファイルから数値文字列の単語コードを取得し、取得した数値文字列の単語コードを比較部３３５に出力する。比較部３３５は、検索範囲の最大値と最小値とに対応する単語コードと、ファイルリード部３３４によって取得された単語コードとを比較する。比較部３３５は、ファイルリード部３３４によって取得された単語コードが検索範囲内に該当するか否かを判定する。そして、比較部３３５は、比較対象の数値文字列が検索範囲内であると判定した場合に、比較対象の数値文字列に対応する単語コードをファイルライト部３３６に出力する。

【００９９】

ここで、数値文字列に割り当てられる単語コードについて説明する。図８の例を用いて説明したように、０から１６，３８３までの整数に対して３バイトの単語コードが割り当てられ、１６，３８４から１，０４８，５７５までの整数に対して４バイトの単語コードが割り当てられる。さらに、１，０４８，５７６から１，０７３，７４１，８２３までの整数に対して５バイトの単語コードが割り当てられる。図８の例に示すｃ１－３、ｃ２－３およびｃ３－３は、半角／全角の別、カンマの有無、小数点の有無、正の値／負の値の別などを示す情報ビットである。すなわち、単語コードの末尾４ビットが情報ビットに該当する。

【０１００】

例えば、整数「0」「1」「2」「3」「4」「5」…に対して、それぞれ３バイトの単語コード「B00000h」「B00010h」「B00020h」「B00030h」「B00040h」…が割り当てられる。すなわち、若い整数から順番に単語コードを割当てる場合、各整数に対して「B00000h」から昇順に単語コードが割り当てられる。これにより、単語コードの状態で数値文字列同士を大小比較することができる。例えば、比較部３３５は、整数「1」に割当てられた単

語コード「B00010h」と、整数「3」に割当てられた単語コード「B00030h」とを比較することで、整数「3」の方が大きいと判定することができる。

【０１０１】

半角／全角の別、カンマの有無等の表記形式の異なる数値文字列同士の比較について説明する。例えば、数値文字列「1200」には、単語コード「B04B00h」が割当てられ、数値文字列「1,200」には、単語コード「B04B01h」が割当てられ、それぞれ異なる単語コードが割当てられるが、これらは同じ大きさの数値文字列である。比較部３３５は、表記形式の異なる数値文字列に係る単語コード同士を比較するためにマスク処理を実行する。

【０１０２】

マスク処理の具体例について説明する。比較部３３５は、数値文字列「1200」の単語コード「B04B00h」と数値文字列「1,200」の単語コード「B04B01h」とを比較する場合、それぞれの単語コードにマスク処理用のビット列を乗算する。すなわち、比較部３３５は、数値文字列「1200」の単語コード（２進数）「1011 0000 0100 1011 0000 0000」に対して、マスク処理用のビット列「1111 1111 1111 1111 1111 0000」を乗算し、ビット列「1011 0000 0100 1011 0000 0000」を取得する。また、比較部３３５は、数値文字列「1,200」の単語コード（２進数）「1011 0000 0100 1011 0000 0001」に対して、マスク処理用のビット列「1111 1111 1111 1111 1111 0000」を乗算し、ビット列「1011 0000 0100 1011 0000 0000」を取得する。そして、比較部３３５は、取得したビット列同士を比較し、数値文字列「1200」の単語コード「B04B00h」と数値文字列「1,200」の単語コード「B04B01h」とが等しいと判定する。

【０１０３】

このように、比較部３３５は、比較対象の各数値文字列に対し、末尾４ビットが「0」となっているマスク処理用のビット列を乗算することで、表記形式の異なる数値文字列に係る単語コード同士を比較することができる。

【０１０４】

図２０を用いて、比較部３３５でなされる大小比較処理の具体例について説明する。図２０は、数値文字列の大小比較を説明するための図である。表Ｄ１は、検索範囲の最大値および最小値に対応する単語コードを表す。一方、表Ｄ２は、比較対象の数値文字列に対応する単語コードを表す。なお、比較対象の数値文字列とは、ファイルリード部３３４によって圧縮ファイルから取得された圧縮符号に係る各単語コードである。

【０１０５】

比較部３３５は、検索範囲の最小値「110」に対応する単語コード「B006E0h」と、比較対象の数値文字列「115」に対応する単語コード「B00730h」とを比較する。比較部３３５は、１６進数で表記される単語コードを２進数に置き換える。すなわち、比較部３３５は、検索範囲の最小値「110」に対応する単語コード「B006E0h」を、「1011」「0000」「0000」「0110」「1111」「0000」に置き換える。また、比較部３３５は、比較対象の数値文字列「115」に対応する単語コード「B00730h」を、「1011」「0000」「0000」「0111」「0011」「0000」に置き換える。そして、比較部３３５は、２進数に置き換えられた単語コードを４ビット毎に比較し、比較対象の数値文字列「115」が検索範囲の最小値よりも大きいと判定する。

【０１０６】

比較部３３５は、検索範囲の最大値「125」に対応する単語コード「B007D0h」と、比較対象の数値文字列「115」に対応する単語コード「B00730h」とを比較する。比較部３３５は、検索範囲の最大値「125」に対応する単語コード「B007D0h」を、「1011」「0000」「0000」「1000」「0010」「0000」に置き換える。そして、比較部３３５は、２進数に置き換えられた４ビット毎に比較し、比較対象の数値文字列「115」が検索範囲の最大値よりも小さいと判定する。

【０１０７】

比較部３３５は、比較対象の数値文字列が検索範囲の最小値よりも大きく、検索範囲の最大値よりも小さいので、比較対象の数値文字列が検索範囲内に含まれると判定する。次

いで、比較部３３５は、比較対象の数値文字列「115」に対応する単語コード「B00730h」をファイルライト部３３６に出力する。

【０１０８】

ファイルライト部３３６は、検索範囲内の数値文字列の単語コードを数値文字列に変換し、検索結果として出力する処理部である。ファイルライト部３３６は、比較部３３５から出力された単語コード「B00730h」を数値文字列「115」に変換して、所定の形式でモニタ、プリンタ等の出力媒体に表示する。例えば、ファイルライト部３３６は、数値文字列「115」と共に、数値文字列「115」が含まれるファイル名、数値文字列「115」が含まれるページ数および行数を出力媒体に表示する。

【０１０９】

（実施例３の大小比較処理の全体の流れ）

次に、実施例３の大小比較処理の全体の流れについて説明する。図２１は、実施例３の大小比較処理の全体の流れを示す図である。図２１の例のように、情報処理装置３００は、前処理をおこなう（ステップＳ５０）。例えば、情報処理装置３００は、前処理として大小比較処理を行うための作業領域を確保する。ファイル選択部３３２は、ビットマップ型全文インデックス３２３の情報に基づいてオープンする圧縮ファイルを選択する（ステップＳ５１）。比較部３３５は、ファイル選択部３３２によって選択された圧縮ファイルから検索範囲内に含まれる単語コードを抽出する（ステップＳ５２）。ファイルライト部３３６は、検索範囲内に含まれる各単語コードをそれぞれ数値文字列に変換し、変換された各数値文字列を所定の形式で表示媒体に出力することで比較結果を表示する（ステップＳ５３）。なお、表示媒体は、例えば、モニタ、プリンタ等である。

【０１１０】

（実施例３の圧縮ファイル選択処理の流れ）

次に、実施例３の圧縮ファイル選択処理の流れについて説明する。図２２は、実施例３の圧縮ファイル選択処理の流れを示す図である。図２２に示される処理は、ステップＳ５１に対応する。図２２の例のように、情報処理装置１００は、前処理をおこなう（ステップＳ６０）。例えば、情報処理装置１００は、前処理として選択結果マップを生成するための作業領域を確保する。

【０１１１】

ファイル選択部３３２は、検索範囲受付部３３１が受け付けた検索範囲の最大値および最小値に対応する単語コードを取得する（ステップＳ６１）。ファイル選択部３３２は、取得した検索範囲の最大値および最小値に対応する単語コードに基づいて、検索範囲内に属するビットマップをビットマップ型全文インデックス３２３から抽出する（ステップＳ６２）。ファイル選択部３３２は、ビットマップ型全文インデックス３２３から抽出した複数のビットマップ間でＯＲ演算をおこなうことで選択結果マップを生成する（ステップＳ６３）。ファイル選択部３３２は、生成された選択結果マップに基づいてオープンする圧縮ファイルを選択する。

【０１１２】

（実施例３の数値文字列の単語コード抽出処理の流れ）

次に、実施例３の数値文字列の単語コード抽出処理の流れについて説明する。図２３は、実施例３の数値文字列の単語コード抽出処理の流れを示す図である。図２３に示される処理は、ステップＳ５２に対応する。図２３の例のように、情報処理装置３００は、前処理をおこなう（ステップＳ７０）。例えば、情報処理装置３００は、前処理として、伸長辞書３２４を記憶するための領域や大小比較処理を行うための作業領域を確保する。

【０１１３】

伸長辞書生成部３３３は、伸長辞書を生成する（ステップＳ７１）。比較部３３５は、検索範囲の最大値および最小値に対応する数値文字列の単語コードを取得する（ステップＳ７２）。例えば、比較部３３５は、伸長辞書３２４から検索範囲の最大値および最小値に対応する数値文字列の単語コードを取得する。ファイルリード部３３４は、選択結果マップに基づいて選択された圧縮ファイルを読み出す（ステップＳ７３）。ファイルリード

部３３４は、選択された圧縮ファイルから順次１６ビットの数値文字列に係る圧縮符号を取得する（ステップＳ７４）。比較部３３５は、圧縮ファイルから取得した圧縮符号に対応する数値文字列の単語コードを伸長辞書３２４から取得する（ステップＳ７５）。

【０１１４】

比較部３３５は、数値文字列の単語コードが検索範囲内に該当するか否かを判定する（ステップＳ７６）。比較部３３５は、数値文字列の単語コードが検索範囲内に該当する場合（ステップＳ７６Yes）、数値文字列の単語コードを伸長文字（数値文字列）に変換し（ステップＳ７７）、比較結果として伸長文字をモニタ等の表示媒体に出力する（ステップＳ７８）。一方、比較部３３５は、数値文字列の単語コードが検索範囲内に該当しない場合（ステップＳ７６No）、ステップＳ７９の処理に移行する。

【０１１５】

ファイルリード部３３４は、ファイルの終端まで至ったか否かを判定する（ステップＳ７９）。ファイルリード部３３４は、ファイルの終端まで至った場合（ステップＳ７９Yes）、処理を終了させる。一方、ファイルリード部３３４は、ファイルの途中である場合（ステップＳ７９No）、ステップＳ７４の処理に戻る。

【０１１６】

（実施例３の効果）

処理部３３０は、数値を示す、少なくとも１つの数字を含む数値文字列とともに数値の大きさの順番に応じて割り当てられるコードと圧縮符号とを対応付けて記憶する辞書を用いて、圧縮ファイルから抽出した圧縮符号をコードに変換する。処理部３３０は、変換したコードと、数値文字列の検索範囲に対応する１以上のコードとを比較することで、変換したコードが検索範囲内に含まれるか否かを判定する。処理部３３０は、検索範囲内に含まれると判定した場合に、変換したコードを数値文字列に伸長して表示する。これにより、処理部３３０は、全ての単語コードを数値文字列に伸長しなくても、検索範囲に含まれる単語コードのみを数値文字列に伸長すればよいので、数値文字列の検索を高速化できる。

【０１１７】

処理部３３０は、各数値文字列を含む圧縮ファイルに関する情報を有するインデックスを用いて、検索範囲内の数値文字列を含む圧縮ファイルを特定する。これにより、検索範囲内の数値文字列を有する圧縮ファイルを特定し、オープンする圧縮ファイルを絞り込むので、複数の圧縮ファイルがある場合でも数値文字列検索を高速化できる。

【０１１８】

（実施例１～３に関連する他の態様）

以下、上述の実施形態における変形例の一部を説明する。下記の変形例のみでなく、本発明の本旨を逸脱しない範囲の設計変更は適宜行われうる。

【０１１９】

また、圧縮処理の対象は、ファイル内のデータ以外にも、システムから出力される監視メッセージなどでもよい。例えば、バッファに順次格納される監視メッセージを上述の圧縮処理により圧縮し、ログファイルとして格納するなどの処理が行なわれる。また、例えば、データベース内のページ単位に圧縮が行なわれてもよいし、複数のページをまとめた単位で圧縮が行なわれてもよい。

【０１２０】

また、実施例１に示した処理手順、制御手順、具体的名称、各種のデータやパラメータを含む情報については、特記する場合を除いて任意に変更することができる。

【０１２１】

（情報処理装置のハードウェア構成）

図２４は、実施例１～３の情報処理装置のハードウェア構成を示す図である。図２４の例が示すように、コンピュータ４００は、各種演算処理を実行するＣＰＵ４０１と、ユーザからのデータ入力を受け付ける入力装置４０２と、モニタ４０３とを有する。また、コンピュータ４００は、記憶媒体からプログラム等を読み取る媒体読取装置４０４と、他の

装置と接続するためのインターフェース装置４０５と、他の装置と無線により接続するための無線通信装置４０６とを有する。また、コンピュータ４００は、各種情報を一時記憶するＲＡＭ４０７と、ハードディスク装置４０８とを有する。また、各装置４０１～４０８は、バス４０９に接続される。

【０１２２】

ハードディスク装置４０８には、例えば図６に示したファイルリード部１１１、圧縮符号付与部１１２およびファイルライト部１１３の各処理部と同様の機能を有する情報処理プログラムが記憶される。さらに、ハードディスク装置４０８には、ファイルリード部１１１、圧縮符号付与部１１２およびファイルライト部１１３の各処理部と同様の機能を有する情報処理プログラムが記憶される。また、ハードディスク装置４０８には、情報処理プログラムを実現するための各種データが記憶される。

【０１２３】

ＣＰＵ４０１は、ハードディスク装置４０８に記憶された各プログラムを読み出して、ＲＡＭ４０７に展開して実行することで各種の処理を行う。これらのプログラムは、コンピュータ４００を、例えば図６に示したファイルリード部１１１、圧縮符号付与部１１２およびファイルライト部１１３として機能させることができる。さらに、これらのプログラムは、コンピュータ４００を、ファイルリード部１１１、圧縮符号付与部１１２およびファイルライト部１１３として機能させることができる。

【０１２４】

なお、上記の情報処理プログラムは、必ずしもハードディスク装置４０８に記憶されている必要はない。例えば、コンピュータ４００が読み取り可能な記憶媒体に記憶されたプログラムを、コンピュータ４００が読み出して実行するようにしてもよい。コンピュータ４００が読み取り可能な記憶媒体は、例えば、ＣＤ－ＲＯＭやＤＶＤディスク、ＵＳＢ（Universal Serial Bus）メモリ等の可搬型記録媒体、フラッシュメモリ等の半導体メモリ、ハードディスクドライブ等が対応する。また、公衆回線、インターネット、ＬＡＮ（Local Area Network）等に接続された装置にこのプログラムを記憶させておき、コンピュータ４００がこれらからプログラムを読み出して実行するようにしてもよい。

【０１２５】

図２５は、コンピュータで動作するプログラムの構成例を示す図である。コンピュータ４００において、図２４に示すハードウェア群２６（４０１～４０９）の制御を行なうＯＳ（オペレーティング・システム）２７が動作する。ＯＳ２７に従った手順でＣＰＵ４０１が動作して、ハードウェア群２６の制御・管理が行なわれることにより、アプリケーションプログラム２９やミドルウェア２８に従った処理がハードウェア群２６で実行される。さらに、コンピュータ４００において、ミドルウェア２８またはアプリケーションプログラム２９が、ＲＡＭ４０７に読み出されてＣＰＵ４０１により実行される。

【０１２６】

ＣＰＵ４０１により圧縮機能が呼び出された場合、ミドルウェア２８またはアプリケーションプログラム２９の少なくとも一部に基づく処理を行なうことで、（それらの処理をＯＳ２７に基づいてハードウェア群２６を制御して）圧縮部１１０の機能が実現される。圧縮機能は、それぞれアプリケーションプログラム２９自体に含まれてもよいし、アプリケーションプログラム２９に従って呼び出されることで実行されるミドルウェア２８の一部であってもよい。

【０１２７】

アプリケーションプログラム２９（またはミドルウェア２８）の圧縮機能により得られる圧縮ファイルは、部分的に伸張することも可能である。圧縮ファイルの途中を伸張する場合には、伸張対象の部分までの圧縮データの伸張処理が抑制されるため、ＣＰＵ４０１の負荷が抑制される。また、伸張対象の圧縮データを部分的にＲＡＭ４０７上に展開するので、ワークエリアも削減される。

【０１２８】

図２６は、実施形態のシステムにおける装置の構成例を示す図である。図２６のシステ

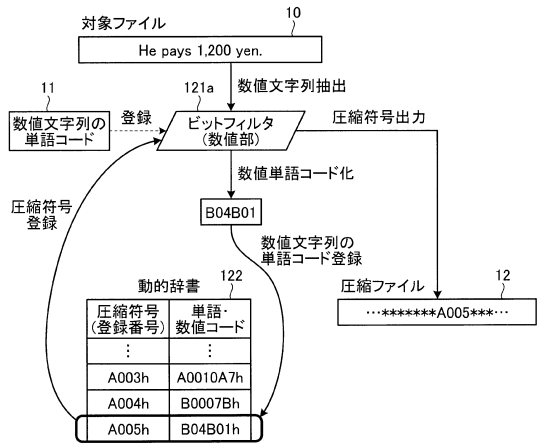
ムは、コンピュータ４００ａ、コンピュータ４００ｂ、基地局３０およびネットワーク４０を含む。コンピュータ４００ａは、無線または有線の少なくとも一方により、コンピュータ４００ｂと接続されたネットワーク４０に接続している。

【符号の説明】

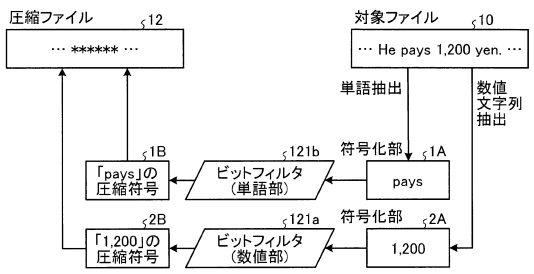
【０１２９】

１００	情報処理装置
１１０	圧縮部
１１１	ファイルリード部
１１２	圧縮符号付与部
１２０	記憶部
１２１	ビットフィルタ
１２１ａ	数値部
１２１ｂ	単語部
１２２	動的辞書

【図１】



【図２】



【図３Ａ】

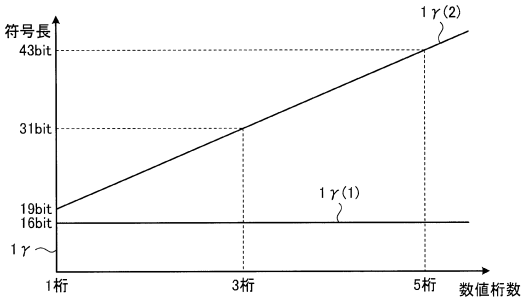
参考例1の符号長を説明するための図

【図３Ｂ】

実施例1の符号長を説明するための図

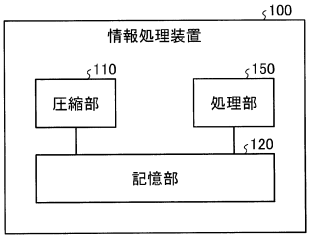
【図４】

参考例1および実施例1の符号長の比較を説明するための図



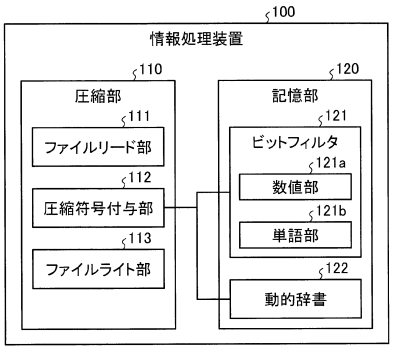
【図５】

情報処理装置のシステム構成の例を説明するための図



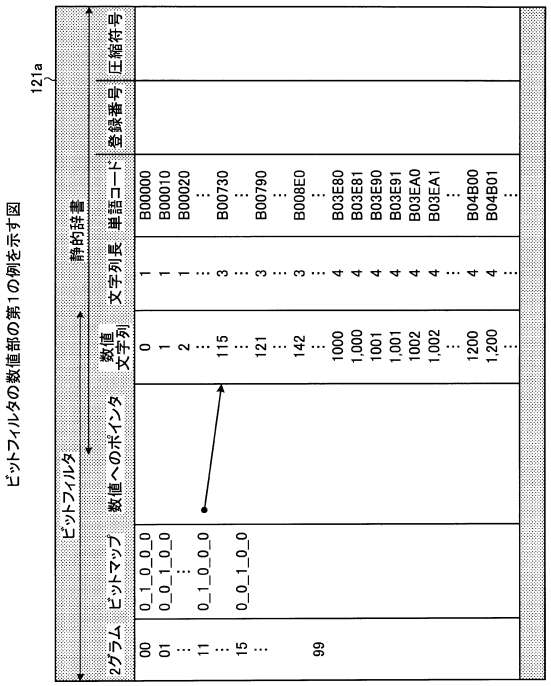
【図６】

実施例1の圧縮処理に係るシステム構成の例を示す図

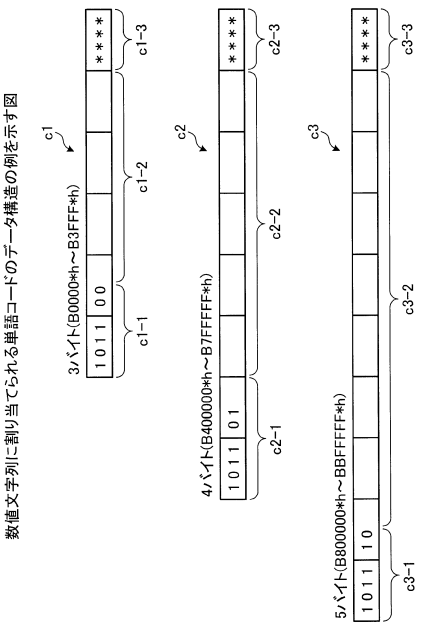


【図７】

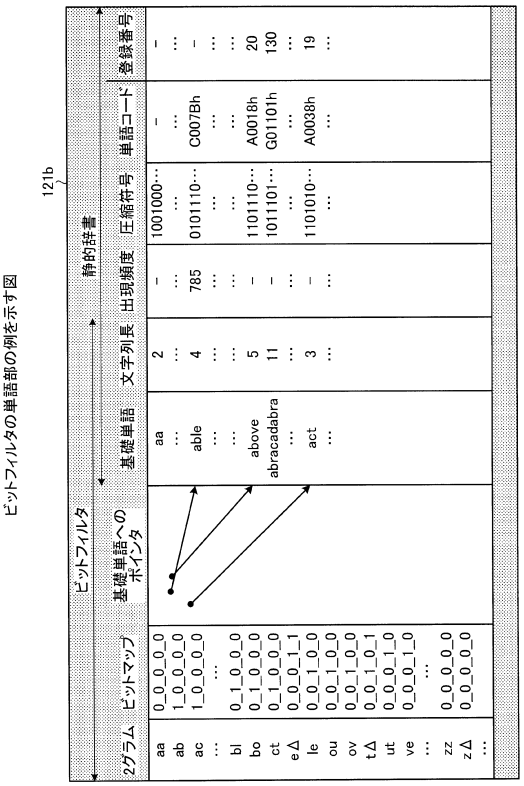
ビットフィルタの数値部の第1の例を示す図



【図８】



【図９】



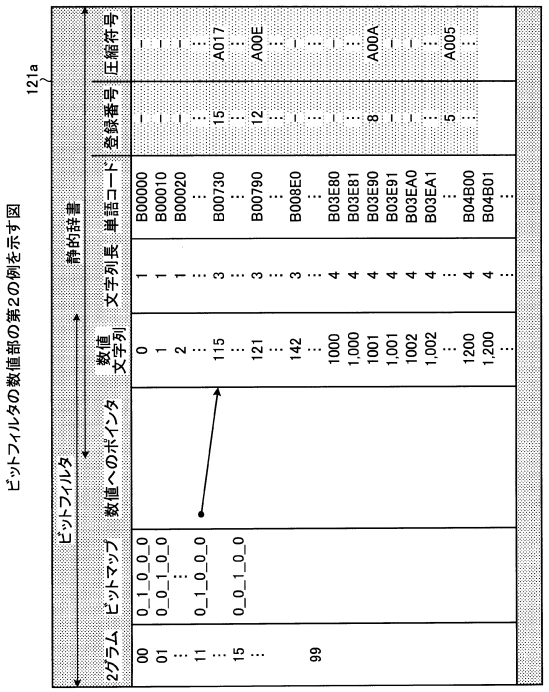
【図１０】

動的辞書の一例を示す図

122

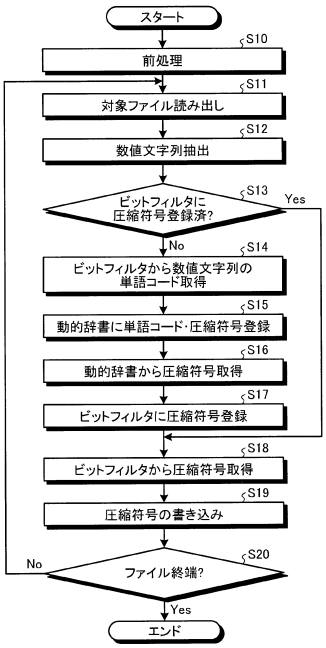
オフセット	圧縮符号	コード
0x0000	A000h	c(order)
0x0004	A001h	c(box)
0x0007	A002h	c(1000)
0x0010	A003h	c(him)
0x0013	A004h	c(pays)
0x0018	A005h	c(1,200)
…	…	…
0x002F	A016h	c(121)
…	…	…
0x0035	A018h	c(able)
…	…	…
0x003F	A025h	c(Kataoka)
…	…	…
0x0046	A036h	c(1024)
…	…	…

【図１１】



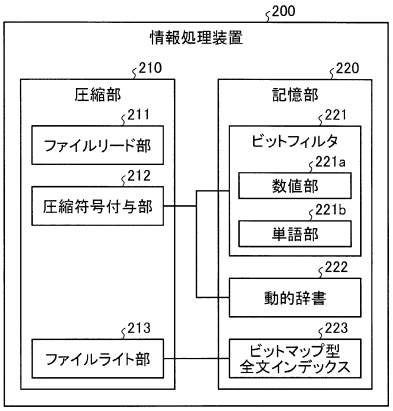
【図１２】

実施例1の圧縮処理の流れの例を示す図



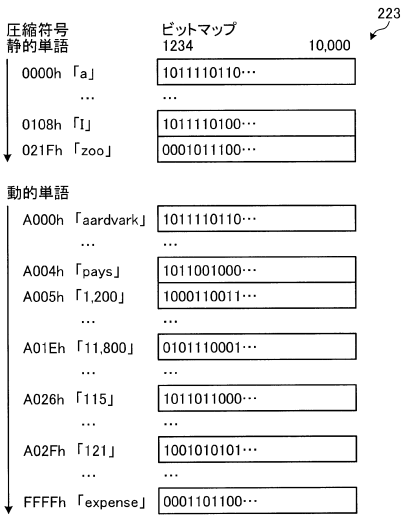
【図１３】

実施例2の圧縮処理に係るシステム構成の例を示す図



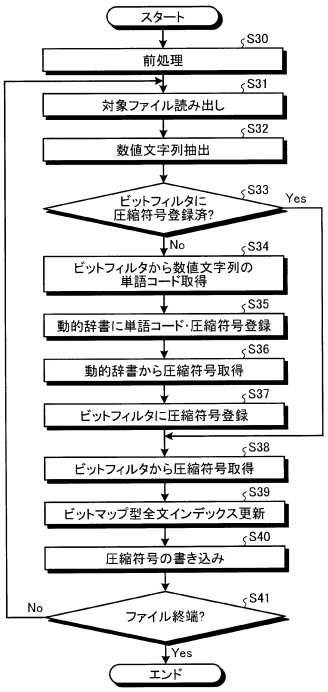
【図１４】

ビットマップ型全文インデックスの一例を示す図



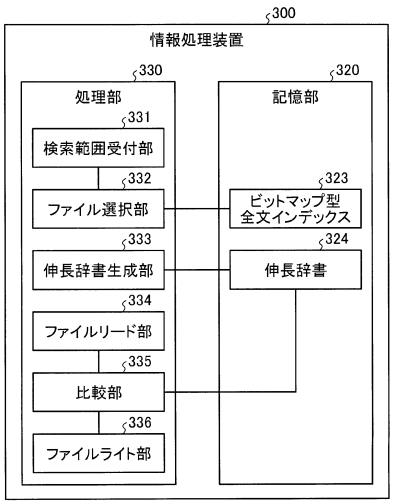
【図１５】

実施例2の圧縮処理の流れを示す図



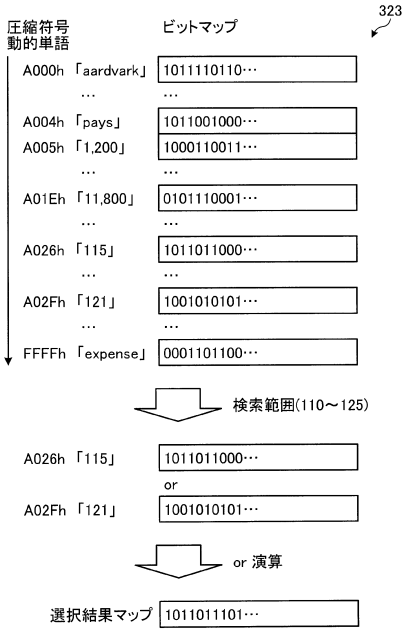
【図１６】

実施例3の大小比較処理に係るシステム構成の一例を示す図



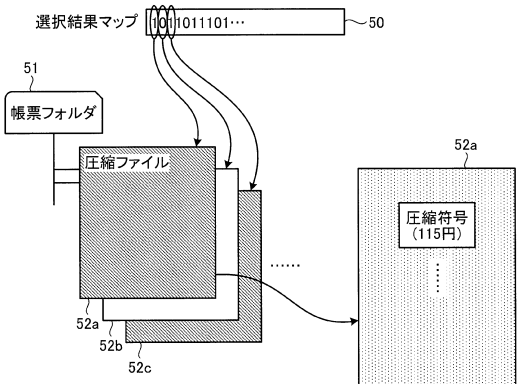
【図１７】

圧縮ファイル選択の処理の流れの例を示す第1の図



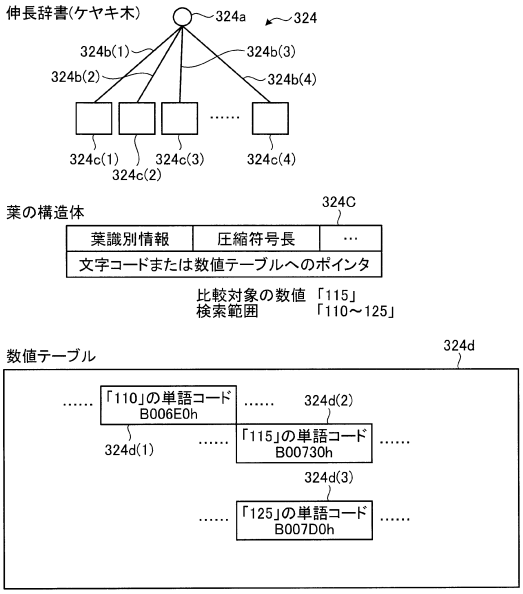
【図１８】

圧縮ファイル選択の処理の流れの例を示す第2の図

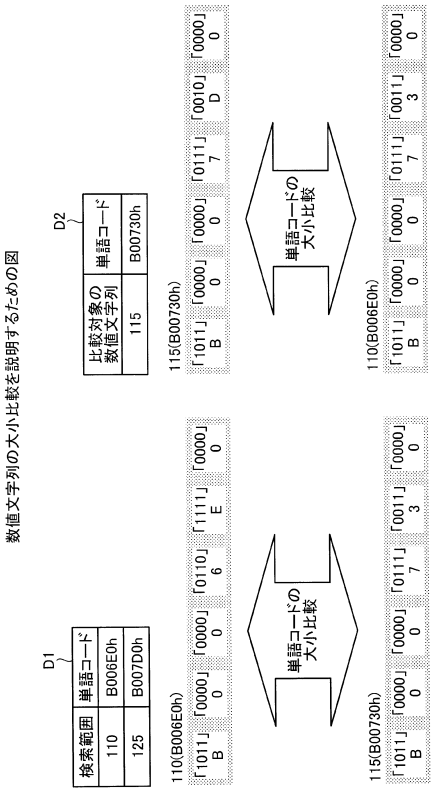


【図１９】

伸長辞書の構造を説明するための図

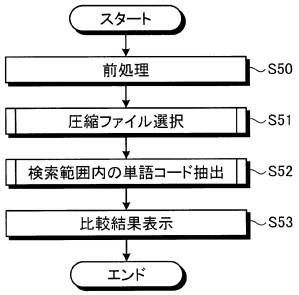


【図２０】



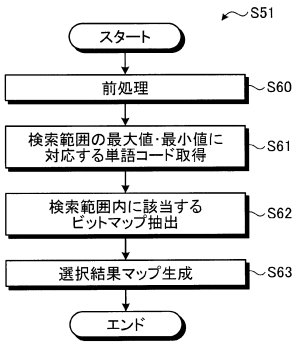
【図２１】

実施例3の大小比較処理の全体の流れを示す図



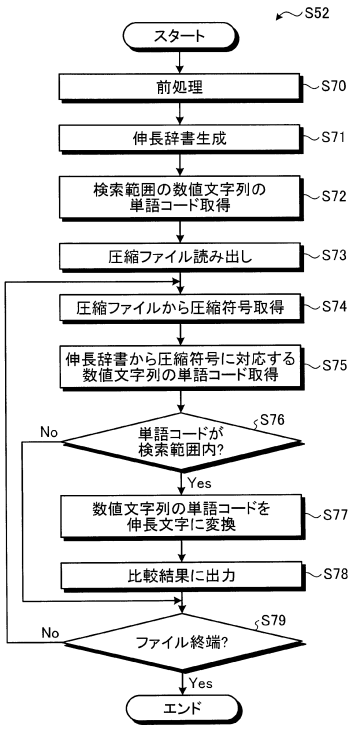
【図２２】

実施例3の圧縮ファイル選択処理の流れを示す図



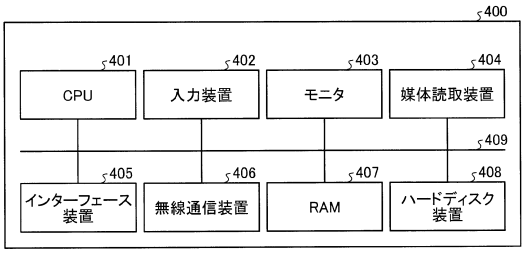
【図２３】

実施例3の数値文字列の単語コード抽出処理の流れを示す図



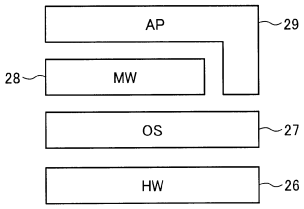
【図２４】

実施例１～３の情報処理装置のハードウェア構成を示す図



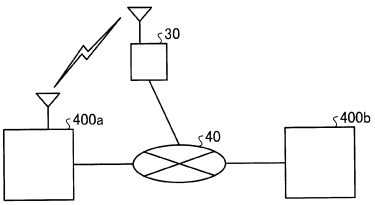
【図２５】

コンピュータで動作するプログラムの構成例を示す図



【図２６】

実施形態のシステムにおける装置の構成例を示す図



フロントページの続き

審査官　北村　智彦

(56)参考文献　国際公開第２００８／０４７４３２（WO，Ａ１）
特開平０５－１２０３５８（ＪＰ，Ａ）
特開２０１１－０１９０１１（ＪＰ，Ａ）
特開２００８－２８７４１２（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０３Ｍ　　３／００－９／００
ＩＥＥＥ　Ｘｐｌｏｒｅ