(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第6252489号
(P6252489)

(45)発行日　平成29年12月27日(2017.12.27)　(24)登録日　平成29年12月8日(2017.12.8)

(51)Int.Cl.		F I		
G06F 12/00	(2006.01)	G06F	12/00	511A
G06F 5/00	(2006.01)	G06F	5/00	

請求項の数 11　(全 33 頁)

(21)出願番号	特願2014-552753 (P2014-552753)
(86)(22)出願日	平成24年12月19日 (2012.12.19)
(86)国際出願番号	PCT/JP2012/008114
(87)国際公開番号	WO2014/097353
(87)国際公開日	平成26年6月26日 (2014.6.26)
審査請求日	平成27年9月10日 (2015.9.10)
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　100105142
弁理士　下田　憲次
(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　鈴木　泰裕
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　山本　貢嗣
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】圧縮装置、圧縮方法、圧縮プログラム、伸張装置、伸張方法、伸張プログラム、および圧縮伸張システム

(57)【特許請求の範囲】

【請求項１】

コンピュータに、

対象データから文字単位でロードされた文字列より、固定長符号化辞書を用いて固定長符号列を生成し、

生成された前記固定長符号列に対しスライド窓を用いた最長一致探索を用いて、前記固定長符号列を圧縮する、

処理を実行させ、

前記最長一致探索は、前記固定長符号列中の圧縮対象である第１固定長符号列において先頭から順に取り出される特定固定長符号のそれぞれに対し、前記特定固定長符号が前記スライド窓を構成する固定長符号列において格納されている各位置を示すビット列である第１ビット列と、前記第１固定長符号列において前記特定固定長符号よりも前に取り出された第２固定長符号列が前記スライド窓を構成する固定長符号列において格納されている各位置を示すビット列である第２ビット列とに基づいて、前記第１ビット列と前記第２ビット列との少なくとも一方を前記第２固定長符号列のサイズに応じてビットシフトして論理積演算を行なうことにより得られる第３ビット列を生成することにより実行される、

ことを特徴とする圧縮プログラム。

【請求項２】

前記生成する処理は、第１記憶領域に格納された前記文字列から前記固定長符号列を生成して第２記憶領域に格納する処理を含み、

前記圧縮する処理は、前記第２記憶領域に格納された前記固定長符号列を用いて圧縮する処理である、
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】
前記圧縮する処理は、
前記第２記憶領域に格納された固定長符号列が存在しない場合、前記格納する処理により前記固定長符号列が前記第２記憶領域に格納された後、前記第２記憶領域に格納された前記固定長符号列を用いて圧縮し、
前記第２記憶領域に格納された固定長符号列が存在する場合、前記第２記憶領域に格納された前記固定長符号列を用いて圧縮する処理である、
ことを特徴とする請求項２に記載の圧縮プログラム。
【請求項４】
前記対象データは、ＡＳＣＩＩ及びＵＴＦ－８のような複数の文字種を含むデータであり、
前記固定長符号列における各固定長符号は、前記複数の文字種を固定長符号化したものである、
ことを特徴とする請求項１～３のいずれか１項に記載の圧縮プログラム。
【請求項５】
前記対象データは、ＡＳＣＩＩのみを含むデータであり、
前記固定長符号列は、前記対象データに含まれる文字または単語を符号化した符号列である、
ことを特徴とする請求項１～４のいずれか１項に記載の圧縮プログラム。
【請求項６】
コンピュータに、
対象データから文字単位でロードされた文字列より、固定長符号化辞書を用いて固定長符号列を生成し、
生成された前記固定長符号列に対しスライド窓を用いた最長一致探索を用いて、前記固定長符号列を圧縮する、
処理を実行させ、
前記最長一致探索は、前記固定長符号列中の圧縮対象である第１固定長符号列において先頭から順に取り出される特定固定長符号のそれぞれに対し、前記特定固定長符号が前記スライド窓を構成する固定長符号列において格納されている各位置を示すビット列である第１ビット列と、前記第１固定長符号列において前記特定固定長符号よりも前に取り出された第２固定長符号列が前記スライド窓を構成する固定長符号列において格納されている各位置を示すビット列である第２ビット列とに基づいて、前記第１ビット列と前記第２ビット列との少なくとも一方を前記第２固定長符号列のサイズに応じてビットシフトして論理積演算を行なうことにより得られる第３ビット列を生成することにより実行される、
処理を実行させることを特徴とする圧縮方法。
【請求項７】
対象データから文字単位でロードされた文字列より、固定長符号化辞書を用いて固定長符号列を生成する変換部と、
生成された前記固定長符号列に対しスライド窓を用いた最長一致探索を用いて、前記固定長符号列を圧縮する生成部と、
を含み、
前記最長一致探索は、前記固定長符号列中の圧縮対象である第１固定長符号列において先頭から順に取り出される特定固定長符号のそれぞれに対し、前記特定固定長符号が前記スライド窓を構成する固定長符号列において格納されている各位置を示すビット列である第１ビット列と、前記第１固定長符号列において前記特定固定長符号よりも前に取り出された第２固定長符号列が前記スライド窓を構成する固定長符号列において格納されている各位置を示すビット列である第２ビット列とに基づいて、前記第１ビット列と前記第２ビ

ット列との少なくとも一方を前記第２固定長符号列のサイズに応じてビットシフトして論理積演算を行なうことにより得られる第３ビット列を生成することにより実行される、
ことを特徴とする圧縮装置。
【請求項８】
コンピュータに、
記憶領域内の位置を示す圧縮符号に基づく前記記憶領域の参照により固定長符号を取得し、
取得した前記固定長符号に基づいて、前記記憶領域の更新を行なうとともに、取得した前記固定長符号を固定長符号化辞書に基づき復号化する、
ことを実行させ、
前記圧縮符号は、
対象データから文字単位でロードされた文字列より、前記固定長符号化辞書を用いて固定長符号列を生成し、
生成された前記固定長符号列に対しスライド窓を用いた最長一致探索を用いて、前記固定長符号列を圧縮する、処理によって生成され、
前記最長一致探索は、前記固定長符号列中の圧縮対象である第１固定長符号列において先頭から順に取り出される特定固定長符号のそれぞれに対し、前記特定固定長符号が前記スライド窓を構成する固定長符号列において格納されている各位置を示すビット列である第１ビット列と、前記第１固定長符号列において前記特定固定長符号よりも前に取り出された第２固定長符号列が前記スライド窓を構成する固定長符号列において格納されている各位置を示すビット列である第２ビット列とに基づいて、前記第１ビット列と前記第２ビット列との少なくとも一方を前記第２固定長符号列のサイズに応じてビットシフトして論理積演算を行なうことにより得られる第３ビット列を生成することにより実行される、
ことを特徴とする伸張プログラム。
【請求項９】
コンピュータに、
記憶領域内の位置を示す圧縮符号に基づく前記記憶領域の参照により固定長符号を取得し、
取得した前記固定長符号に基づいて、前記記憶領域の更新を行なうとともに、取得した前記固定長符号を固定長符号化辞書に基づき復号化する、
ことを実行させ、
前記圧縮符号は、
対象データから文字単位でロードされた文字列より、前記固定長符号化辞書を用いて固定長符号列を生成し、
生成された前記固定長符号列に対しスライド窓を用いた最長一致探索を用いて、前記固定長符号列を圧縮する、処理によって生成され、
前記最長一致探索は、前記固定長符号列中の圧縮対象である第１固定長符号列において先頭から順に取り出される特定固定長符号のそれぞれに対し、前記特定固定長符号が前記スライド窓を構成する固定長符号列において格納されている各位置を示すビット列である第１ビット列と、前記第１固定長符号列において前記特定固定長符号よりも前に取り出された第２固定長符号列が前記スライド窓を構成する固定長符号列において格納されている各位置を示すビット列である第２ビット列とに基づいて、前記第１ビット列と前記第２ビット列との少なくとも一方を前記第２固定長符号列のサイズに応じてビットシフトして論理積演算を行なうことにより得られる第３ビット列を生成することにより実行される、
ことを特徴とする伸張方法。
【請求項１０】
記憶領域内の位置を示す圧縮符号に基づく前記記憶領域の参照により固定長符号を取得する取得部と、
取得した前記固定長符号に基づいて、前記記憶領域の更新を行なう更新部と、
取得した前記固定長符号を固定長符号化辞書に基づき復号化する変換部と、

を含み、
前記圧縮符号は、
対象データから文字単位でロードされた文字列より、前記固定長符号化辞書を用いて固定長符号列を生成し、
生成された前記固定長符号列に対しスライド窓を用いた最長一致探索を用いて、前記固定長符号列を圧縮する、処理によって生成され、
前記最長一致探索は、前記固定長符号列中の圧縮対象である第１固定長符号列において先頭から順に取り出される特定固定長符号のそれぞれに対し、前記特定固定長符号が前記スライド窓を構成する固定長符号列において格納されている各位置を示すビット列である第１ビット列と、前記第１固定長符号列において前記特定固定長符号よりも前に取り出された第２固定長符号列が前記スライド窓を構成する固定長符号列において格納されている各位置を示すビット列である第２ビット列とに基づいて、前記第１ビット列と前記第２ビット列との少なくとも一方を前記第２固定長符号列のサイズに応じてビットシフトして論理積演算を行なうことにより得られる第３ビット列を生成することにより実行される、
ことを特徴とする伸張装置。
【請求項１１】
対象データから文字単位でロードされた文字列より、固定長符号化辞書を用いて固定長符号列を生成する変換部と、
生成された前記固定長符号列に対しスライド窓を用いた最長一致探索を用いて、前記固定長符号列を圧縮して圧縮符号を生成する生成部と、
を含む第１のコンピュータと、
前記圧縮符号と前記固定長符号化辞書とに基づき、前記文字列を復元する復元部、
を含む第２のコンピュータと、
を含み、
前記最長一致探索は、前記固定長符号列中の圧縮対象である第１固定長符号列において先頭から順に取り出される特定固定長符号のそれぞれに対し、前記特定固定長符号が前記スライド窓を構成する固定長符号列において格納されている各位置を示すビット列である第１ビット列と、前記第１固定長符号列において前記特定固定長符号よりも前に取り出された第２固定長符号列が前記スライド窓を構成する固定長符号列において格納されている各位置を示すビット列である第２ビット列とに基づいて、前記第１ビット列と前記第２ビット列との少なくとも一方を前記第２固定長符号列のサイズに応じてビットシフトして論理積演算を行なうことにより得られる第３ビット列を生成することにより実行される、
ことを特徴とする圧縮伸張システム。
【発明の詳細な説明】
【技術分野】
【０００１】
データの圧縮技術または伸張技術の少なくとも一方に関する。
【背景技術】
【０００２】
ＬＺ７７と呼ばれる圧縮アルゴリズムがある。ＬＺ７７においては、処理対象のデータよりも先に出現し、且つ処理対象のデータと同一であるデータの位置と長さに基づいて圧縮符号が生成される。先に出現した同一データの探索は、先に出現した各データとの照合処理により行なわれる。照合処理では、処理対象のデータと先に出現したデータとを所定のデータ単位ごとに比較が行なわれる。例えば、所定のデータ単位が１バイトであると、符号化対象のデータと先に出現したデータとの照合処理が１バイトごとに行なわれる。
【先行技術文献】
【特許文献】
【０００３】
【特許文献１】特開平０８－２３４９５９号公報
【発明の概要】

【解決しようとする課題】
【０００４】
しかしながら、圧縮対象のデータを構成するデータ単位の長さが一定でないこともありうる。文書データにおいて、例えば、単一の文字を表現するバイト数が複数種類用いられる文字コード系が存在する。ＵＴＦ－８などにおいては、１バイトで表現される文字（例えば英数字など）もあれば、３バイトで表現される文字（例えば、漢字第１種の一部、第２種漢字およびかな文字など）、４バイトで表現される文字（例えば漢字第３・第４水準の一部など）も存在する。ＵＴＦ－８などの圧縮対象のデータ内に含まれるデータのデータ単位が複数種類用いられている圧縮対象のデータに対しても、圧縮対象のデータを構成する実際のデータ単位（例えば複数バイト）と異なるデータ単位（例えば１バイト）での照合処理が行なわれる。
【０００５】
本発明の一側面においては、データを構成するデータ単位が複数種類用いられるデータに対する圧縮処理において行なわれる照合処理を効率化させることを目的とする。
【課題を解決するための手段】
【０００６】
一態様によれば、コンピュータに、対象データから文字単位でロードされた文字列より、固定長符号化辞書を用いて固定長符号列を生成し、生成された前記固定長符号列に対しスライド窓を用いた最長一致探索を用いて、前記固定長符号列を圧縮する、処理を実行させ、前記最長一致探索は、前記固定長符号列中の圧縮対象である第１固定長符号列において先頭から順に取り出される特定固定長符号のそれぞれに対し、前記特定固定長符号が前記スライド窓を構成する固定長符号列において格納されている各位置を示すビット列である第１ビット列と、前記第１固定長符号列において前記特定固定長符号よりも前に取り出された第２固定長符号列が前記スライド窓を構成する固定長符号列において格納されている各位置を示すビット列である第２ビット列とに基づいて、前記第１ビット列と前記第２ビット列との少なくとも一方を前記第２固定長符号列のサイズに応じてビットシフトして論理積演算を行なうことにより得られる第３ビット列を生成することにより実行される、圧縮方法が用いられる。
【０００７】
一態様によれば、コンピュータに、記憶領域内の位置を示す圧縮符号に基づく前記記憶領域の参照により固定長符号を取得し、取得した前記固定長符号に基づいて、前記記憶領域の更新を行なうとともに、取得した前記固定長符号を固定長符号化辞書に基づき復号化する、ことを実行させる伸張方法が用いられる。前記圧縮符号は、対象データから文字単位でロードされた文字列より、前記固定長符号化辞書を用いて固定長符号列を生成し、生成された前記固定長符号列に対しスライド窓を用いた最長一致探索を用いて、前記固定長符号列を圧縮する、処理によって生成されている。前記最長一致探索は、前記固定長符号列中の圧縮対象である第１固定長符号列において先頭から順に取り出される特定固定長符号のそれぞれに対し、前記特定固定長符号が前記スライド窓を構成する固定長符号列において格納されている各位置を示すビット列である第１ビット列と、前記第１固定長符号列において前記特定固定長符号よりも前に取り出された第２固定長符号列が前記スライド窓を構成する固定長符号列において格納されている各位置を示すビット列である第２ビット列とに基づいて、前記第１ビット列と前記第２ビット列との少なくとも一方を前記第２固定長符号列のサイズに応じてビットシフトして論理積演算を行なうことにより得られる第３ビット列を生成することにより実行される。
【０００８】
一態様によれば、圧縮伸張システムが、対象データから文字単位でロードされた文字列より、固定長符号化辞書を用いて固定長符号列を生成する変換部と、生成された前記固定長符号列に対しスライド窓を用いた最長一致探索を用いて、前記固定長符号列を圧縮して圧縮符号を生成する生成部と、を含む第１のコンピュータと、前記圧縮符号と前記固定長符号化辞書とに基づき、前記文字列を復元する復元部、を含む第２のコンピュータと、を

含む。前記最長一致探索は、前記固定長符号列中の圧縮対象である第１固定長符号列において先頭から順に取り出される特定固定長符号のそれぞれに対し、前記特定固定長符号が前記スライド窓を構成する固定長符号列において格納されている各位置を示すビット列である第１ビット列と、前記第１固定長符号列において前記特定固定長符号よりも前に取り出された第２固定長符号列が前記スライド窓を構成する固定長符号列において格納されている各位置を示すビット列である第２ビット列とに基づいて、前記第１ビット列と前記第２ビット列との少なくとも一方を前記第２固定長符号列のサイズに応じてビットシフトして論理積演算を行なうことにより得られる第３ビット列を生成することにより実行される。

【発明の効果】

【０００９】

一側面によれば、圧縮処理において、圧縮対象のデータを構成するデータ単位と異なるデータ単位での照合処理が行なわれることを抑制することができる。

【図面の簡単な説明】

【００１０】

【図１】図１は、ＬＺ７７を利用した圧縮処理の流れを示す。

【図２】図２は、ＬＺ７７を利用した伸張処理の流れを示す。

【図３】図３は、ＵＴＦ－８におけるコードの割り当てを示す。

【図４】図４は、圧縮処理の例を示す。

【図５】図５は、符号化辞書Ｄ１の例を示す。

【図６】図６は、符号化辞書Ｄ２の例を示す。

【図７】図７は、伸張処理の例を示す。

【図８】図８は、機能構成例を示す。

【図９】図９は、位置情報テーブルＴ１の例を示す。

【図１０】図１０は、圧縮処理の手順例を示す。

【図１１】図１１は、最長一致固定長符号列の探索処理の手順例を示す。

【図１２】図１２は、固定長符号の取得処理の手順例を示す。

【図１３】図１３は、圧縮データの生成・書込み処理の手順例を示す。

【図１４】図１４は、記憶領域Ａ２の更新処理の手順例を示す。

【図１５】図１５は、記憶領域Ａ４の更新処理の手順例を示す。

【図１６】図１６は、位置情報テーブルＴ２の例を示す。

【図１７】図１７は、伸張処理の手順例を示す。

【図１８】図１８は、記憶領域Ｂ２の更新処理の手順例を示す。

【図１９】図１９は、コンピュータ１のハードウェア構成例を示す。

【図２０】図２０は、コンピュータ１で動作するプログラムの構成例を示す。

【図２１】図２１は、実施形態のシステムにおける装置の構成例を示す。

【図２２】図２２は、圧縮対象のデータを構成するデータ単位と異なるデータ単位での照合処理例を示す。

【図２３】図２３は、圧縮対象のデータを構成するデータ単位と異なるデータ単位での照合処理例を示す。

【図２４】図２４は、Ｓ３０１からＳ３０３の処理の例を示す。

【図２５】図２５は、符号化辞書Ｄ１のインデックス例を示す。

【図２６】図２６は、最長一致符号列の探索処理の変形例を示す。

【図２７】図２７は、最長一致符号列の探索処理の手順例を示す。

【発明を実施するための形態】

【００１１】

図１は、ＬＺ７７を利用した圧縮処理の流れを示す。まず、記憶領域Ａ１、記憶領域Ａ２および記憶領域Ａ３が例えばメモリ内に確保される。図１に示すファイルＦ１内のコンテンツ部分のデータは、記憶領域Ａ１にロードされる。記憶領域Ａ１は、例えば符号化部などと呼ばれる。ファイルＦ１は、「・・・１ｓｔ　ｈｏｒｓｅ・・・２ｎｄ　ｈｏｒｓ

ｅ・・・３ｒｄ　ｈｏｒｓｅ・・・」というデータが含まれる（「・・・」は不特定な文字列である）。記憶領域Ａ１にロードされたデータに基づいて、圧縮データの生成処理（後述）が行なわれる。また、圧縮データの生成処理が行なわれたデータは、記憶領域Ａ１から記憶領域Ａ２にコピーされる。記憶領域Ａ２は、例えば参照部と呼ばれる。圧縮データは、記憶領域Ａ１にロードされたデータと記憶領域Ａ２内のデータとの照合処理の結果に応じて生成される。生成された圧縮データは順次記憶領域Ａ３に格納され、記憶領域Ａ３に格納された圧縮データに基づいて圧縮ファイルＦ２が生成される。また、図１において、記憶領域Ａ１およびＡ２内のデータは模式的に示されている。

【００１２】

図１に示される「１ｓｔ　ｈｏｒｓｅ・・・」の「ｈ」以降が処理対象のデータである場合を例に圧縮データｄ１の生成を説明する。まず、記憶領域Ａ２内で「ｈｏｒｓｅ・・・」の最長一致データが探索される（図１に示す「照合」）。図１の例において、処理対象のデータの先頭のデータである「ｈ」と一致するデータが記憶領域Ａ２に存在しない。処理対象のデータと一致するデータが記憶領域Ａ２に存在しない場合には、処理対象のデータの先頭データをハフマン符号化／復号化アルゴリズムにより符号化して得られるハフマン符号を含む圧縮データｄ１が生成される。圧縮データとしてハフマン符号化を用いることは、あくまで一例であり、他の圧縮アルゴリズムが用いられてもよいし、先頭データそのものである非圧縮データが用いられてもよい。圧縮データｄ１には、最長一致データに基づく圧縮データでない旨を示す識別子（図１の例において「０」）が含まれる。

【００１３】

図１に示される「２ｎｄ　ｈｏｒｓｅ・・・」の「ｈ」以降が処理対象のデータである場合を例に圧縮データｄ２の生成を説明する。まず、記憶領域Ａ２内で「ｈｏｒｓｅ・・・」の最長一致データが探索される（図１に示す「照合」）。図１の例では、「１ｓｔ　ｈｏｒｓｅ・・・」が記憶領域Ａ２に存在するので、例えば、処理対象のデータの「ｈｏｒｓｅ」と記憶領域Ａ２内の「１ｓｔ　ｈｏｒｓｅ・・・」の「ｈｏｒｓｅ」とが一致する。例えば、記憶領域Ａ２内の一致データ「ｈｏｒｓｅ」が、記憶領域Ａ２内で処理対象データと最も長く一致するデータ（最長一致データ）である場合には、最長一致データの記憶領域Ａ２内での位置と、最長一致データのデータ長に基づき圧縮データｄ２が生成される。圧縮データｄ２には、最長一致データに基づく圧縮データである旨を示す識別子（図１の例において「１」）が含まれる。

【００１４】

図１に示される「３ｒｄ　ｈｏｒｓｅ・・・」の「ｈ」以降が処理対象のデータである場合を例に圧縮データｄ３の生成を説明する。まず、記憶領域Ａ２内で「ｈｏｒｓｅ・・・」の最長一致データが探索される（図１に示す「照合」）。図１の例では、「１ｓｔ　ｈｏｒｓｅ・・・２ｎｄ　ｈｏｒｓｅ」が記憶領域Ａ２に存在するので、例えば、処理対象のデータの「ｈｏｒｓｅ」と記憶領域Ａ２内の「１ｓｔ　ｈｏｒｓｅ」および「２ｎｄ　ｈｏｒｓｅ」の「ｈｏｒｓｅ」とが一致する。例えば、記憶領域Ａ２内の「１ｓｔ　ｈｏｒｓｅ」または「２ｎｄ　ｈｏｒｓｅ」のいずれか「ｈｏｒｓｅ」が最長一致データである場合に、最長一致データの記憶領域Ａ２内での位置と、最長一致データのデータ長に基づき圧縮データｄ３が生成される。圧縮データｄ３には、最長一致データに基づく圧縮データである旨を示す識別子（図１の例において「１」）が含まれる。

【００１５】

生成された圧縮データｄ１～ｄ３は、記憶領域Ａ３に記憶され、圧縮ファイルＦ２の生成処理により、圧縮ファイルＦ２に含まれる。

【００１６】

図２は、ＬＺ７７を利用した伸張処理の流れを示す。伸張処理においては、圧縮ファイルＦ２内の圧縮データをメモリ（記憶領域Ｂ１）にロードし、ロードされた圧縮データの識別子に応じて伸張データの生成処理を行なう。図２の「＊」は圧縮されたデータであることを示す。記憶領域Ｂ１は、例えば符号化部などと呼ばれる。最長一致データに基づく圧縮データでない旨を示す識別子（図１の例において「０」）を含む圧縮データ（図２に

おける圧縮データｄ１など）を読み出した場合には、ハフマン符号化／復号化アルゴリズムに従った復号処理により、伸張データが生成される。生成された伸張データは、記憶領域Ｂ２および記憶領域Ｂ３の双方に格納される。記憶領域Ｂ２は、例えば参照部などと呼ばれる。

【００１７】

一方、最長一致データに基づく圧縮データである旨を示す識別子（図１の例において「１」）を含む圧縮データ（図２における圧縮データｄ２および圧縮データｄ３など）を読み出した場合には、圧縮符号に示される記憶領域Ｂ２内のデータが伸張データとなる。識別子が最長一致データに基づく圧縮データである旨を示す場合も、生成された伸張データは、記憶領域Ｂ２および記憶領域Ｂ３の双方に格納される。

【００１８】

伸張データを記憶領域Ｂ２に格納することにより、記憶領域Ｂ２が圧縮符号の生成処理が行なわれる際の記憶領域Ａ２と同じ状態にすることができ、そのため圧縮符号に基づいて圧縮する前と同じデータが取得される。記憶領域Ｂ３に格納された伸張データに基づいて伸張ファイルＦ３が生成される。

【００１９】

図３は、ＵＴＦ－８におけるコードの割り当てを示す。ＵＴＦ－８においては、上述の通り、１～４バイトの文字コードが用いられる。文字コードの長さに応じて、文字コードの値の範囲が定められている。

【００２０】

１バイトの文字コードは、０ｘ００～０ｘ７Ｆの値で表現される。そのため、２進数表記では「０ＸＸＸＸＸＸＸ」となり、先頭のビットが「０」となる（「Ｘ」には「０」か「１」かのいずれかの値である）。２バイトの文字コードは、１バイト目が０ｘＣ２～０ｘＤＦの値であり（０ｘＣ０および０ｘＣ１は、例えば制御コードに用いられる）、２バイト目に０ｘ８０～０ｘＢＦの値が用いられる。すなわち、２バイトの文字コードの１バイト目は「１１０ＹＹＹＹＸ」であり、２バイト目は「１０ＸＸＸＸＸＸ」である（「Ｙ」は連続する「Ｙ」のうちいずれか少なくとも１つが「１」であることを示す）。３バイトの文字コードは、１バイト目が０ｘＥ０～０ｘＥＦの値であり、２バイト目及び３バイト目は０ｘ８０～０ｘＢＦの値が用いられる。すなわち、３バイトの文字コードの１バイト目は「１１１０ＹＹＹＹ」であり、２バイト目は「１０ＹＸＸＸＸＸ」であり、３バイト目は「１０ＸＸＸＸＸＸ」である。４バイトの文字コードは、１バイト目が０ｘＦ０～０ｘＦ７の値であり、２～４バイト目は０ｘ８０～０ｘＢＦの値が用いられる。すなわち、４バイト文字コードの１バイト目は「１１１１０ＹＹＹ」であり、２バイト目は「１０ＹＹＸＸＸＸ」であり、３バイト目及び４バイト目はそれぞれ「１０ＸＸＸＸＸＸ」である。

【００２１】

ＵＴＦ－８のコード割り当てでは、２バイト以上の文字コードにおいて、１バイト目のデータと２バイト目以降のデータとは異なる値となる。図１における圧縮処理において、例えば、記憶領域Ａ１内の３バイトの文字コードの１バイト目のデータと、記憶領域Ａ２内の各データと照合が行なわれる。すると、記憶領域Ａ２には、例えば、３バイトの文字コードの２バイト目のデータも、３バイト目のデータも含まれる。２バイト以上の文字コードにおいて、１バイト目のデータと２バイト目以降のデータとは異なる値となるＵＴＦ－８などの文字コード体系では、異なる値であることが明らかであるにも関わらず、１バイト目のデータと２バイト目以降のデータについての照合処理が行なわれてしまう。

【００２２】

ＬＺ７７を利用した圧縮（例えばＺＩＰなどによる圧縮）は、圧縮対象のデータ間での照合結果が得られるデータであれば適用され得る。ＺＩＰなどは、例えば、文書データや画像データなど異なる種別のデータに対しても汎用的に用いられる。種別を選ばず適用可能であるために、特定の種別のデータを対象とする改善が試みられにくい状況であった。しかしながら、特定の文字コード体系におけるデータ間の照合処理に対して敢えて詳細な

手順を追尾することにより、上述のように異なる値になることが明らかなデータ間においても照合処理が行なわれることが明らかとなった。

【００２３】

上述のように、文字コードのデータ単位よりも細かいデータ単位で照合処理が行なわれることにより、不要な照合処理が発生してしまう。本実施形態においては、ＵＴＦ－８などの文字コードのサイズが複数種類存在する文字コード系を用いたデータに対して、文字コードに対応したデータ単位の管理を行ない、さらに管理されたデータ単位ごとに照合が行なわれる。

【００２４】

また、異なる３バイト文字に対して、文字コードの境界を無視した圧縮符号化が行われることとなる。例えば「十一」（０ｘＥ２ＢＣ９８Ｅ３８６９２）と「十二」（０ｘＥ２ＢＣ９８Ｅ３８６）との照合により、０ｘＥ２ＢＣ９８Ｅ３８６（５バイト）が一致データ列として抽出され、圧縮符号が割り当てられることとなる。その場合には、文字コードの残りの部分（「十一」については、０ｘ９２）から照合の対象となり、文字コードの境界とずれたままの照合処理（「泣き別れ」）が発生してしまう。それにより圧縮率の低下が見込まれる。

【００２５】

図４は、圧縮処理の例を示す。まず、記憶領域Ａ１、記憶領域Ａ２、記憶領域Ａ３および記憶領域Ａ４が例えばメモリ内に確保される。図４に示すファイルＦ１内のコンテンツ部分のデータは、記憶領域Ａ１にロードされる。記憶領域Ａ１は、例えば符号化部などと呼ばれる。ファイルＦ１は、「・・・１ｓｔ　ｈｏｒｓｅ・・・２ｎｄ　ｈｏｒｓｅ・・・３ｒｄ　ｈｏｒｓｅ・・・」というデータが含まれる（「・・・」は不特定な文字列である）。

【００２６】

記憶領域Ａ１にロードされたデータは、符号化辞書Ｄ１に基づいて固定長符号に変換される。変換で得られた固定長符号に基づいて圧縮データの生成処理が行なわれる。また、圧縮データが生成された固定長符号は記憶領域Ａ２に格納される。記憶領域Ａ２は、例えば参照部と呼ばれる。圧縮データは、変換により得られた固定長符号と記憶領域Ａ２に格納された固定長符号との照合処理の結果に応じて生成される。生成された圧縮データは順次記憶領域Ａ３に格納され、記憶領域Ａ３に格納された圧縮データに基づいて圧縮ファイルＦ２が生成される。また、図４において、記憶領域Ａ１およびＡ２内のデータは模式的に示されている。

【００２７】

図４の例においては、記憶領域Ａ１から文字コードＬ１が読み出され、読み出された文字コードＬ１に対応する固定長符号Ｍ１が符号化辞書Ｄ１から読み出される。読み出された固定長符号Ｍ１は、記憶領域Ａ４に格納される。記憶領域Ａ４に格納された固定長符号Ｍ１に基づいて、記憶領域Ａ２内に格納された固定長符号に対して順次照合処理が行なわれる。記憶領域Ａ４に格納された固定長符号Ｍ１と一致する固定長符号Ｎ１が記憶領域Ａ２内に存在する場合には、さらに記憶領域Ａ１から文字コードＬ２が読み出され、読み出された文字コードＬ２に対応する固定長符号Ｍ２が符号化辞書Ｄ１から読み出されて記憶領域Ａ４に格納される。さらに、記憶領域Ａ２内で固定長符号Ｎ１に後続する固定長符号Ｎ２が固定長符号Ｍ２と一致するかが判定される。一致すれば、さらに文字コードが記憶領域Ａ１から読み出され同様の手順が繰り返される。上述の手順は、一致しない固定長符号が得られるか、連続して一致する固定長符号の数が下限値（例えば、所定の符号数）Ｌｍｉｎを超えるまで繰り返される。記憶領域Ａ２全体に渡って同様の処理が行なわれ、記憶領域Ａ２の中から最も長く一致する固定長符号の列（最長一致固定長符号列）が抽出される。

【００２８】

最長一致固定長符号列が下限値Ｌｍｉｎ以上の長さである場合には、圧縮データｄ１１が生成される。圧縮データｄ１１は、最長一致固定長符号列に基づく圧縮符号である旨を

示す識別子（図４の例では「１」）と、最長一致固定長符号列の長さ（例えば最長一致固定長符号列に含まれる固定長符号の数）と最長一致固定長符号列の位置とを示す圧縮符号とを含む。最長一致固定長符号列の位置は、記憶領域Ａ２の更新位置から固定長符号何個分離れた位置であるかを示す符号の個数などで示される。さらに、記憶領域Ａ４に格納された固定長符号列が記憶領域Ａ２に書き込まれる。記憶領域Ａ２の全領域に固定長符号が書き込まれている場合には、記憶領域Ａ２に格納された固定長符号のうち、記憶領域Ａ２に最も先に格納された固定長符号に、記憶領域Ａ４に格納されている固定長符号列が上書きされる。

【００２９】

最長一致固定長符号列が下限値Ｌｍｉｎよりも短い場合には、圧縮データｄ１２が生成される。圧縮データｄ１２は、最長一致固定長符号列に基づく圧縮符号でない旨を示す識別子（図４の例では「０」）と、固定長符号Ｍ１とを含む。さらに、固定長符号Ｍ１が記憶領域Ａ２に書き込まれる。記憶領域Ａ２の全領域に固定長符号が書き込まれている場合には、記憶領域Ａ２に格納された固定長符号のうち記憶領域Ａ２に最も先に格納された固定長符号に、固定長符号Ｍ１が上書きされる。

【００３０】

上述の手順で圧縮データが生成され、生成されるたびに記憶領域Ａ３に書き込まれる。記憶領域Ａ３に格納された圧縮データに基づいて圧縮ファイルＦ２が生成される。符号化辞書Ｄ１も圧縮ファイルＦ２に含まれるか、または、符号化辞書Ｄ１は他の方法によって圧縮ファイルＦ２を伸張するコンピュータに渡される。さらに詳細な圧縮処理の手順については後述される。

【００３１】

図５は、符号化辞書Ｄ１の例を示す。符号化辞書Ｄ１は、文字コードと固定長符号との対応関係を示す。図５に示す符号化辞書Ｄ１は、日本語の文書を対象とした符号化辞書の例である。図５の例においては、固定長符号の符号長は１２ビットである。また、図５の例においては、各文字コードに対して４バイトずつ格納領域が設けられ、文字コードが格納された位置を示す情報が固定長符号として用いられる。例えば、符号化辞書Ｄ１内の先頭に「ＮＵＬ」のコードが格納されているため、「ＮＵＬ」のコード（０ｘ００）に対応する固定長符号が「０ｘ０００」とする。また、例えば「ａ」の文字コード（０ｘ４１）は符号化辞書Ｄ１の先頭から、４バイト×３２（１６進数表記では０ｘ０２０）の位置に存在するため、「ａ」の文字コードに対応する固定長符号は「０ｘ０２０」となる。

【００３２】

符号化辞書Ｄ１では、各文字コードに対して固定長の符号が割り当てられる。符号長をｍビットとすると、固定長符号が割り当てられる文字コードの数は２のｍ乗となる。図５の例においては符号長が１２ビットなので、４０９６種類の文字コードに符号長が割り当てられる。ファイルＦ１に使用される文字コード系の全文字コードに対して固定長符号を割り当ててもよいし、一部の文字コードに圧縮符号を割り当てることとしてもよい。一部の文字コードに固定長符号を割り当てる場合の制御は後述される。

【００３３】

図６は、符号化辞書Ｄ２の例を示す。符号化辞書Ｄ２は、文字コードまたは文字コード列と固定長符号との対応関係を示す。図６に示す符号化辞書Ｄ２は、英語の文書を対象とした符号化辞書の例である。図６の例においては、固定長符号の符号長は１２ビットである。また、図６の例においては、文字コードまたは文字コード列に対して所定長の格納領域が設けられ、文字コードまたは文字コード列が格納された位置を示す情報が固定長符号として用いられる。

【００３４】

図６に示す符号化辞書Ｄ２においても、例えば、「ＮＵＬ」や「ａ」に対して、図５に示す符号化辞書と同様の固定長符号が割り当てられる。符号化辞書Ｄ２においては、さらに、基礎的な英単語に対しても固定長符号が割り当てられる。図６に示す通り、英単語「ｏｎｅ」に対し、例えば固定長符号「０ｘ１００」が割り当てられる。

【００３５】
図４に示す圧縮処理において、記憶領域Ａ４に格納する固定長符号を生成するにあたって、記憶領域Ａ１の読出し位置に存在するデータ列と合致するデータ列に対応する固定長符号が符号化辞書Ｄ２（図４における符号化辞書Ｄ１に対応）から抽出され、記憶領域Ａ４に格納される。この際、例えば、記憶領域Ａ１の読出し位置に「ａｒｅ」という単語が存在すると、固定長符号０ｘ０２０（「ａ」の文字コード）も固定長符号０ｘ１８０（「ａｒｅ」の文字コード）も抽出されるが、例えば、固定長符号０ｘ０００～０ｘ０ＦＦよりも０ｘ１００～０ｘＦＦＦが優先されると予め定めておく。

【００３６】
英語の文書には、基礎的な単語を高頻度で使用する傾向がある。英語の文書中に含まれる英単語の約半分が、約千語の基礎的な単語で占められる。そのため、図６に示す符号化辞書Ｄ２のように１２ビットの固定長を割り当てられる英単語群であれば、英語の文書をほぼ表現しうる。図６に示す符号化辞書Ｄ２を用いることで、複数回の１バイト単位の照合処理で処理されるデータ量が、一度の照合で処理される。さらに、その一度の照合においても照合対象のデータサイズは固定長符号の符号長にとどめられる。そのため、図６に示す符号化辞書Ｄ２を用いて符号化された固定長符号化同士の照合を行なうことにより、圧縮速度が向上する。

【００３７】
図７は、伸張処理の例を示す。まず、記憶領域Ｂ１、記憶領域Ｂ２および記憶領域Ｂ３が例えばメモリ内に確保される。図７に示す圧縮ファイルＦ２内の圧縮データは、記憶領域Ｂ１にロードされる。記憶領域Ｂ１は、例えば符号化部などと呼ばれる。また、圧縮ファイルＦ２からメモリに符号化辞書Ｄ１がロードされる。上述したように、符号化辞書Ｄ１は圧縮ファイルＦ２内に含まれていなくても、圧縮に用いた符号化辞書Ｄ１が事前に保持されていてもよい。

【００３８】
記憶領域Ｂ１にロードされた圧縮データは順次読み出される。読み出された圧縮データは、圧縮データに含まれる識別子に応じた伸張処理が行なわれる。識別子が、最長一致固定長符号列に基づく圧縮符号でない旨を示す識別子（図７の例では「０」）である場合の圧縮データの例として、圧縮データｄ１２が図７に例示される。圧縮データｄ１２に含まれる固定長符号Ｍ１は、符号化辞書Ｄ１に基づいて復号化される。また、圧縮データｄ１２内の固定長符号Ｍ１が記憶領域Ｂ２の更新位置に書き込まれる。符号化辞書Ｄ１に基づく復号化により得られた文字コードｄ２２は、記憶領域Ｂ３に書き込まれる。

【００３９】
識別子が、最長一致固定長符号列に基づく圧縮符号である旨を示す識別子（図７の例では「１」）である場合の圧縮データの例として、圧縮データｄ１１が図７に例示される。圧縮データｄ１１に含まれる最長一致固定長符号列の長さおよび位置の情報に基づいて、記憶領域Ｂ２から固定長符号列ｄ２１（例えば、固定長符号列Ｍ１～Ｍｎ）が読み出される。固定長符号列ｄ２１が読み出されると、固定長符号列ｄ２１が記憶領域Ｂ２の更新位置に書き込まれるとともに、符号化辞書Ｄ１を用いて復号化される。復号化により得られた文字コード列ｄ２３（例えば、固定長符号列Ｍ１～Ｍｎに対応する文字コード列Ｌ１～Ｌｎ）は記憶領域Ｂ３に書き込まれる。

【００４０】
記憶領域Ｂ２の更新位置への書込みにおいて、記憶領域Ｂ２の全領域に固定長符号が書き込まれている場合には、記憶領域Ｂ２に格納された固定長符号のうち、記憶領域Ｂ２に最も先に格納された固定長符号に対する上書きにより書込みが行なわれる。

【００４１】
記憶領域Ｂ３に順次書き込まれるデータ（文字コード）に基づいて、伸張ファイルＦ３が生成される。さらに詳細な伸張処理の手順については後述される。

【００４２】
図８は、機能構成例を示す。本実施形態の処理を実行するコンピュータ１は、記憶部１

３を含み、さらに、圧縮部１１と伸張部１２との少なくとも一方を含む。圧縮部１１は圧縮処理を行ない、伸張部１２は伸張処理を行なう。記憶部１３は、圧縮対象のファイルＦ１や、圧縮処理により得られる圧縮ファイルＦ２や、ファイルＦ２を伸張して得られるファイルＦ３などを格納する。例えば、記憶部１３は、符号化辞書Ｄ１を記憶する。また、記憶部１３は、圧縮部１１や伸張部１２のワークエリアとして用いられる。圧縮部１１は、制御部１１１、照合部１１２、更新部１１３および変換部１１４を含む。伸張部１２は、制御部１２１、参照部１２２、更新部１２３および変換部１２４を含む。

【００４３】

制御部１１１は、照合部１１２および更新部１１３を制御して、圧縮機能を実現させる。また、制御部１１１は、各機能部の処理に用いるデータを保持するため、記憶部１３に記憶領域（例えば、上述の記憶領域Ａ１、記憶領域Ａ２および記憶領域Ａ３）を確保する。制御部１１１は、順次記憶領域Ａ１内の読出し位置のデータを読み出す。変換部１１４は、制御部１１１が読みだしたデータを符号化辞書Ｄ１に基づいて固定長符号に変換する。制御部１１１は変換部１１４により変換された固定長符号を記憶領域Ａ４に格納する。照合部１１２は、記憶領域Ａ４内の固定長符号に基づいて、記憶領域Ａ２内の固定長符号の参照処理を実行する。更新部１１３は、記憶領域Ａ４内の固定長符号に基づいて、記憶領域Ａ２内の固定長符号列を更新する。制御部１１１は、照合部１１２による記憶領域Ａ２内の参照結果に応じて、圧縮データを生成する。圧縮部１１内の各機能部による処理の実行手順については後述する。

【００４４】

制御部１２１は、参照部１２２および更新部１２３を制御して、伸張機能を実現させる。また、制御部１２１は、各機能部の処理に用いるデータを保持するため、記憶部１３に記憶領域（例えば、上述の記憶領域Ｂ１、記憶領域Ｂ２および記憶領域Ｂ３）を確保する。制御部１２１は、記憶領域Ｂ１内の読出し位置の圧縮データを読み出し、読みだした圧縮データに含まれる識別子を判定する。制御部１２１は、識別子が所定の識別子である場合に、参照部１２２に、記憶領域Ｂ２内の固定長符号の参照処理を実行させる。参照部１２２による参照か、記憶領域Ｂ３からの読出しにより固定長符号が得られると、更新部１２３は、得られた固定長符号に基づいて記憶領域Ｂ２の更新を行なう。また、変換部１２４は、得られた固定長符号を符号化辞書Ｄ１に基づいて伸張データに変換する。伸張部１２内の各機能部による処理の実行手順については後述する。

【００４５】

図９は、記憶領域の位置情報の管理に用いられる位置情報テーブルＴ１の例を示す。位置情報テーブルＴ１は、圧縮処理に用いられる各記憶領域（記憶領域Ａ１、記憶領域Ａ２および記憶領域Ａ３など）の記憶部１３における位置の管理に用いられる。位置情報テーブルＴ１には、ファイルＦ１をロードする記憶領域Ａ１の開始位置Ｐ１、終了位置Ｐ２および読出し位置Ｐ３が含まれる。また、位置情報テーブルＴ１には、記憶領域Ａ２の開始位置Ｐ４、終了位置Ｐ５、参照位置Ｐ６および更新位置Ｐ７が含まれる。さらに、位置情報テーブルＴ１には、記憶領域Ａ３の開始位置Ｐ８、終了位置Ｐ９および書込み位置Ｐ１０が含まれる。位置情報テーブルＴ１に格納されるそれぞれの位置情報の初期値は、制御部１１１により設定される。各記憶領域の開始位置と終了位置は、圧縮や伸張の対象となるデータ（例えば、ファイル内のヘッダやトレーラ部分を除いた部分）の格納開始位置、終了位置を示す。例えば、読出し位置Ｐ３と開始位置Ｐ１との初期値は同じであり、参照位置Ｐ６および更新位置Ｐ７と開始位置Ｐ４との初期値は同じであり、書込み位置Ｐ１０と開始位置Ｐ８との初期値は同じである。

【００４６】

以下に圧縮処理の手順について説明する。

【００４７】

図１０は、圧縮処理の手順例を示す。まず、コンピュータ１内のオペレーティング・システムやアプリケーションプログラムの動作により圧縮機能が呼び出される（Ｓ１０１）

。圧縮機能が呼び出されると、制御部１１１は、例えば、図１に示す記憶領域Ａ１、記憶領域Ａ２、記憶領域Ａ３、および記憶領域Ａ４の確保や、各記憶領域内の各位置情報（例えば、図９に示す各位置情報）の設定などの前処理を実行する（Ｓ１０２）。

【００４８】

Ｓ１０２の処理を終えると、制御部１１１は、圧縮対象のファイルＦ１のコンテンツ部分を記憶領域Ａ１にロードする（Ｓ１０３）。また、制御部１１１は、ファイルＦ１の終端に基づいて終了位置Ｐ２を設定する。次に、制御部１１１は、最長一致固定長符号列の探索処理を実行する（Ｓ１０４）。

【００４９】

図１１は、最長一致固定長符号列の探索処理の手順例を示す。最長一致固定長符号列の探索処理が開始される（Ｓ２００）と、制御部１１１は、参照位置Ｐ６、一致長Ｌａおよび最長一致位置Ｐａの初期値をセットする（Ｓ２０１）。参照位置Ｐ６及び最長一致位置Ｐａは、開始位置Ｐ４と同じか、もしくは更新位置Ｐ７と同じにセットされる。一致長Ｌａは例えば、「０」などにセットされる。制御部１１１は、さらにカウンタ値ｊを初期値（例えばｊ＝０）にセットする（Ｓ２０２）。

【００５０】

次に、制御部１１１は、記憶領域Ａ４に固定長符号Ｍ（ｊ）が存在するか否かを判定する（Ｓ２０３）。固定長符号Ｍ（ｊ）は、記憶領域Ａ４のｊ番目の位置に格納される固定長符号を意味する。記憶領域Ａ４に固定長符号Ｍ（ｊ）が存在しない場合（Ｓ２０３：Ｎｏ）には、制御部１１１は、変換部１１４に固定長符号Ｍ（ｊ）の取得処理を実行させる（Ｓ２０４）。

【００５１】

図１２は、固定長符号の取得処理の手順例を示す。変換部１１４は、制御部１１１に固定長符号Ｍ（ｊ）の取得処理を指示される（Ｓ３００）と、記憶領域Ａ１の読出し位置Ｐ３に存在する文字コードを読み出す（Ｓ３０１）。１バイト文字なら１バイトのデータが読み出され、２バイト文字なら２バイトのデータが読み出される。次に、変換部１１４は、Ｓ３０１で読みだした文字コードに基づき、Ｓ３０１で読みだした文字コードに対応する固定長符号を符号化辞書Ｄ１から読み出す（Ｓ３０２）。さらに、変換部１１４は、位置情報テーブルに格納された読出し位置Ｐ３を示す情報を更新する（Ｓ３０３）。Ｓ３０３の更新は、Ｓ３０１で変換部１１４が読み出したデータの長さに基づいて行なわれる。例えば、１バイト文字が読み出されれば、読出し位置Ｐ３は１バイト移動される。制御部１１１は、Ｓ３０２で読みだした固定長符号を記憶領域Ａ４のｊ番目の位置に格納する（Ｓ３０４）。上述の通り、記憶領域Ａ４のｊ番目の位置に格納される固定長符号は、固定長符号Ｍ（ｊ）である。変換部１１４は、固定長符号Ｍ（ｊ）を記憶領域に格納すると、固定長符号の取得処理を終了する（Ｓ３０５）。

【００５２】

図１１の説明に戻る。記憶領域Ａ４に固定長符号Ｍ（ｊ）が存在するか（Ｓ２０３：Ｙｅｓ）、Ｓ２０４の固定長符号の取得処理が終了した場合に、制御部１１１は、照合部１１２に照合処理を実行させる（Ｓ２０５）。Ｓ２０５において、照合部１１２は、記憶領域Ａ４に格納された固定長符号Ｍ（ｊ）と、記憶領域Ａ２内で参照位置Ｐ６からカウンタ値ｊに応じて移動させた位置に存在する固定長符号とが一致するか否かを判定する。参照位置Ｐ６からカウンタ値ｊに応じて移動させた位置とは、具体的には、固定長符号の符号長がｍビットであれば、参照位置Ｐ６からｍ×ｊビットずれた位置である。

【００５３】

Ｓ２０５の判定で固定長符号同士が合致した場合（Ｓ２０５：Ｙｅｓ）には、制御部１１１は、カウンタ値ｊのインクリメントを行なう（Ｓ２０６）。次に、制御部１１１は、カウンタ値ｊが上限値Ｌｍａｘに達した（ｊ＝Ｌｍａｘ）か否かを判定する（Ｓ２０７）。上限値Ｌｍａｘは、一致長Ｌａの上限値として設定される値である。一致長Ｌａの表現に用いられるビット数がｍ１と圧縮符号のフォーマットで定められている場合には、例え

ば、２のｍ１乗－１が上限値として設定される。カウンタ値ｊが上限値Ｌｍａｘに達していない場合（Ｓ２０７：Ｎｏ）には、制御部１１１は、Ｓ２０３の処理を実行する。また、カウンタ値ｊが上限値Ｌｍａｘに達している場合（Ｓ２０７：Ｙｅｓ）には、制御部１１１は、一致長Ｌａにカウンタ値ｊを代入し、最長一致位置Ｐａに参照位置Ｐ６を代入する（Ｓ２０８）。図１１のＳ２０８の処理に示す「＝」は代入演算子を示す。

【００５４】

Ｓ２０５の判定で固定長符号同士が合致しない場合（Ｓ２０５：Ｎｏ）には、制御部１１１は、カウンタ値ｊが一致長Ｌａよりも大きいか否かを判定する（Ｓ２０９）。カウンタ値ｊが一致長Ｌａよりも大きい場合（Ｓ２０９：Ｙｅｓ）には、制御部１１１は、一致長Ｌａにカウンタ値ｊを代入し、最長一致位置Ｐａに参照位置Ｐ６を代入する（Ｓ２１０）。図１１のＳ２１０の処理に示す「＝」は代入演算子を示す。カウンタ値ｊが一致長Ｌａ以下であるか（Ｓ２０９：Ｎｏ）、Ｓ２１０の処理が行なわれると、制御部１１１は、記憶領域Ａ２内の参照位置Ｐ６の値をインクリメントする（Ｓ２１１）。具体的には、記憶領域Ａ２に格納される固定長符号の符号長を単位としてインクリメントされ、固定長符号の符号長がｍビットであれば参照位置Ｐ６はｍビット移動される。次に、制御部１１１は、参照位置Ｐ６が記憶領域Ａ２の終点位置Ｐ５に達したか否かを判断する（Ｓ２１２）。Ｓ２１２の判定において、参照位置Ｐ６が終点位置Ｐ５に達していない場合（Ｓ２１２：Ｎｏ）には、制御部１１１は、Ｓ２０２の処理を行なう。

【００５５】

制御部１１１は、Ｓ２０８の処理が行なわれるか、参照位置Ｐ６が終点位置Ｐ５に達している場合（Ｓ２１２：Ｙｅｓ）には、最長一致固定長符号列の探索処理を終了する（Ｓ２１３）。Ｓ１０４の探索処理の結果得られる最長一致固定長符号列は、Ｓ１０４の処理が終了した時点における記憶領域Ａ２内の最長一致位置Ｐａから一致長Ｌａの範囲内の固定長符号列である。一致長Ｌａは一致した符号の数を示すので、固定長符号の符号長がｍビットであれば、最長一致固定長符号列はＬａ×ｍビットの長さとなる。

【００５６】

続いて、制御部１１１は、Ｓ１０４の探索処理の結果に基づいて圧縮データの生成・書込み処理を実行する（Ｓ１０５）。

【００５７】

図１３は、圧縮データの生成・書込み処理の手順例を示す。圧縮データの生成・書込み処理が開始される（Ｓ４００）と、制御部１１１は、一致長Ｌａが下限値Ｌｍｉｎ以上であるか否かを判定する（Ｓ４０１）。下限値Ｌｍｉｎは、一致長Ｌａの下限値として設定される値である。例えば、一致長Ｌａの表現に用いられるビット数がｍ１であり、最長一致位置Ｐａの表現に用いられるビット数がｍ２であると圧縮符号のフォーマットで定められている場合に、Ｌａ×ｍ＜ｍ１＋ｍ２となり得る。その場合には、最長一致固定長符号列を利用した圧縮符号により生成される圧縮データのデータサイズよりも、固定長符号列を用いて生成される圧縮データのデータサイズの方が小さい。そこで、例えば、下限値Ｌｍｉｎ以上の一致長Ｌａであれば、Ｌａ×ｍ≧ｍ１＋ｍ２となるように、下限値Ｌｍｉｎは設定される。下限値Ｌｍｉｎの設定は、他の設定（例えば、ｍ１、ｍ２およびｍなどの値の設定）などに応じて調整される。

【００５８】

一致長Ｌａが下限値Ｌｍｉｎ以上である場合（Ｓ４０１：Ｙｅｓ）には、制御部１１１は、識別子「１」の情報を生成する（Ｓ４０２）。続いて、制御部１１１は、一致長Ｌａを示すｍ１ビットの情報、および最長一致位置Ｐａを示すｍ２ビットの情報を生成する（Ｓ４０３）。Ｓ４０３において、制御部１１１は、例えば、識別子「１」、一致長Ｌａおよび最長一致位置Ｐａの順序で連続する情報を生成する。次に、制御部１１１は、移動量Ｌｃに一致長Ｌａを代入する（Ｓ４０４）。移動量Ｌｃは、圧縮データの生成により、圧縮処理が行なわれた固定長符号の符号数を示す。一致長Ｌａに対応する個数の固定長符号

がＳ４０３により生成される圧縮符号に変換されるので、移動量Ｌｃは一致長Ｌａと同じである。

【００５９】

一致長Ｌａが下限値Ｌｍｉｎよりも短い場合（Ｓ４０１：Ｎｏ）には、制御部１１１は、識別子「０」の情報を生成する（Ｓ４０５）。続いて、制御部１１１は、記憶領域Ａ４に格納された固定長符号Ｍ（０）の読み出しを行なう（Ｓ４０６）。Ｓ４０６において、制御部１１１は、Ｓ４０５で生成した識別子「０」と記憶領域Ａ４から読み出した固定長符号Ｍ（０）を連続させた情報を生成する。さらに、制御部１１１は、移動量Ｌｃに１を代入する（Ｓ４０７）。

【００６０】

ＳＳ４０４またはＳ４０７の処理が行なわれると、制御部１１１は、圧縮データの書込み位置Ｐ１０に圧縮データを書き込む（Ｓ４０８）。圧縮データは、Ｓ４０３またはＳ４０６により生成される情報である。さらに、制御部１１１は、Ｓ４０８で書き込まれる圧縮データの長さに応じて、書込み位置Ｐ１０の更新を行なう。例えば、圧縮データの長さは、Ｓ４０３で生成される圧縮データであれば、１＋ｍ１＋ｍ２ビットである。また、Ｓ４０６で生成される圧縮データの長さは、例えば１＋ｍビットである。Ｓ４０９の処理が行なわれると、制御部１１１は、圧縮データの生成・書込み処理を終了する（Ｓ４１０）。

【００６１】

図１０に戻って説明を続けると、圧縮データが生成され、書込み処理が行なわれると、制御部１１１は、記憶領域Ａ２の更新処理を更新部１１３に実行させる（Ｓ１０６）。

【００６２】

図１４は、記憶領域Ａ２の更新処理の手順例を示す。更新部１１３は、制御部１１１に記憶領域Ａ２の更新処理を指示される（Ｓ５００）と、カウンタ値ｉを初期値（ｉ＝０）にセットする（Ｓ５０１）。次に、更新部１１３は、記憶領域Ａ２の更新位置Ｐ７からカウンタ値ｉに応じて移動された位置に、記憶領域Ａ４に格納された固定長符号Ｍ（ｉ）を書き込む（Ｓ５０２）。具体的には、Ｓ５０２で書き込まれる位置は、固定長符号の符号長ｍとすると、更新位置Ｐ７からｍ×ｉビットずれた位置である。言い換えると、Ｓ５０２で書き込まれる位置は、更新位置Ｐ７を固定長符号の符号長ｍを単位として表現すると、Ｐ７＋ｉで表される位置である。

【００６３】

次に、更新部１１３は、カウンタ値ｉが移動量Ｌｃから１引いた値に達したか否かを判定する（Ｓ５０３）。カウンタ値ｉが移動量Ｌｃから１引いた値になるまで処理が行なわれることによって、記憶領域Ａ４に格納された固定長符号のうち、圧縮符号への変換が行なわれた固定長符号について、記憶領域Ａ２に反映される。

【００６４】

カウンタ値ｉが移動量Ｌｃから１引いた値に達していない場合（Ｓ５０３：Ｎｏ）に、更新部１１３はカウンタ値ｉをインクリメントする（Ｓ５０４）。さらに、Ｓ５０４でインクリメントされたカウンタ値ｉに基づいて、更新位置Ｐ７＋カウンタ値ｉが記憶領域Ａ２の終了位置Ｐ５に達しているかを判断する（Ｓ５０５）。更新位置Ｐ７＋カウンタ値ｉが記憶領域Ａ２の終了位置Ｐ５に達している場合（Ｓ５０５：Ｙｅｓ）は、更新部１１３は、更新位置Ｐ７に、記憶領域Ａ２の開始位置Ｐ４からカウンタ値ｉを引いた値を代入する（Ｓ５０６）。Ｓ５０５およびＳ５０６の処理により、記憶領域Ａ２外にはみ出して固定長符号が格納されることもなく、記憶領域Ａ２が繰り返し使用される。更新位置Ｐ７＋カウンタ値ｉが記憶領域Ａ２の終了位置Ｐ５に達していない場合（Ｓ５０５：Ｎｏ）か、Ｓ５０６の処理が行なわれた場合には、更新部１１３は、Ｓ５０２の処理を行なう。

【００６５】

カウンタ値ｉが移動量Ｌｃから１引いた値に達した場合（Ｓ５０３：Ｙｅｓ）に、更新部１１３は、記憶領域Ａ２の更新位置Ｐ７を更新する（Ｓ５０７）。具体的には、更新位置Ｐ７に、更新位置Ｐ７に移動量Ｌｃを加算した値が代入される。Ｓ５０７の処理を終え

ると、更新部１１３は、記憶領域Ａ２の更新処理を終了する（Ｓ５０８）。
【００６６】
図１０に戻って説明を続けると、制御部１１１は、更新部１１３による記憶領域Ａ２の更新処理が終了すると、更新部１１３に記憶領域Ａ４の更新処理を実行させる（Ｓ１０７）。
【００６７】
図１５は、記憶領域Ａ４の更新処理の手順例を示す。更新部１１３は、制御部１１１に記憶領域Ａ４の更新処理を指示される（Ｓ６００）と、記憶領域Ａ４内の固定長符号Ｍ（０）～Ｍ（Ｌｃ－１）を削除する（Ｓ６０１）。固定長符号Ｍ（０）～Ｍ（Ｌｃ－１）に対応する圧縮データは既に生成され、且つ記憶領域Ａ２にコピーされている。さらに、更新部１１３は、カウンタ値ｋの初期値（ｋ＝０）をセットする（Ｓ６０２）。
【００６８】
次に、更新部１１３は、固定長符号Ｍ（Ｌｃ＋ｋ）が存在するか否かを判断する（Ｓ６０３）。固定長符号Ｍ（Ｌｃ＋ｋ）が存在する場合（Ｓ６０３：Ｙｅｓ）に、更新部１１３は、記憶領域Ａ４内で固定長符号Ｍ（Ｌｃ＋ｋ）をカウンタ値ｋの位置にコピーする（Ｓ６０４）。すなわち、更新部１１３は固定長符号Ｍ（ｋ）を記憶領域Ａ４に格納する。さらに、更新部１１３は、固定長符号Ｍ（Ｌｃ＋ｋ）を削除する（Ｓ６０５）。次に更新部１１３は、カウンタ値ｋをインクリメントする（Ｓ６０６）。Ｓ６０６の処理が行なわれると、更新部１１３は、Ｓ６０３の処理を行なう。また、Ｓ６０３の判定において、固定長符号Ｍ（Ｌｃ＋ｋ）が存在しない場合（Ｓ６０３：Ｎｏ）に、更新部１１３は、記憶領域Ａ４の更新処理を終了する（Ｓ６０７）。
【００６９】
更新部１１３による記憶領域Ａ４の更新処理が終了すると、制御部１１１は、ファイルＦ１の終点まで圧縮処理が終了したか否かを判定する（Ｓ１０８）。Ｓ１０８において、例えば記憶領域Ａ１の読出し位置Ｐ３が、記憶領域Ａ１の終了位置Ｐ２に達したか否かが判定される。ファイルＦ１の終点まで圧縮処理が終了していない場合（Ｓ１０８：Ｎｏ）には、制御部１１１は、Ｓ１０４の処理を行なう。一方、ファイルＦ１の終点まで圧縮処理が終了した場合（Ｓ１０８：Ｙｅｓ）には、制御部１１１は、記憶領域Ａ３内に格納された圧縮データ群に基づいて、圧縮ファイルＦ２の生成処理を行なう（Ｓ１０９）。すなわち、圧縮ファイルＦ２がクローズされ、記憶部１３内に格納される。Ｓ１０９の処理が終了すると、制御部１１１は、圧縮処理を終了する（Ｓ１１０）。Ｓ１１０の処理において、制御部１１１は、例えば、圧縮機能の呼び出し元に対して、圧縮処理の終了通知を行なう。圧縮処理の終了通知には、例えば、圧縮ファイルＦ２の格納先を示す情報などが含まれる。
【００７０】
図１６は、記憶領域の位置情報の管理に用いられる位置情報テーブルＴ２の例を示す。位置情報テーブルＴ２は、伸張処理に用いられる各記憶領域（記憶領域Ｂ１、記憶領域Ｂ２および記憶領域Ｂ３など）の記憶部１３における位置の管理に用いられる。位置情報テーブルＴ２には、圧縮ファイルＦ２がロードされる記憶領域Ｂ１の開始位置Ｑ１、終了位置Ｑ２および読出し位置Ｑ３が含まれる。また、位置情報テーブルＴ２には、記憶領域Ｂ２の開始位置Ｑ４、終了位置Ｑ５、参照位置Ｑ６および更新位置Ｑ７が含まれる。さらに、位置情報テーブルＴ２には、記憶領域Ｂ３の開始位置Ｑ８、終了位置Ｑ９および書込み位置Ｑ１０が含まれる。位置情報テーブルＴ２に格納されるそれぞれの位置情報の初期値は、制御部１１１により設定される。各記憶領域の開始位置と終了位置は、圧縮や伸張の対象となるデータ（例えば、ファイル内のヘッダやトレーラ部分を除いた部分）の格納開始位置、終了位置を示す。例えば、読出し位置Ｑ３と開始位置Ｑ１との初期値は同じであり、参照位置Ｑ６および更新位置Ｑ７の初期値は、開始位置Ｑ４と同じであり、書込み位置Ｑ１０と開始位置Ｑ８との初期値は同じである。
【００７１】
以下に伸張処理の手順について説明する。

【００７２】
図１７は、伸張処理の手順例を示す。まず、コンピュータ１内のオペレーティング・システムやアプリケーションプログラムの動作により伸張機能が呼び出される（Ｓ７００）。伸張機能が呼び出されると、制御部１２１は、例えば、図２に示す記憶領域Ｂ１、記憶領域Ｂ２、記憶領域Ｂ３および記憶領域Ｂ４の確保、各記憶領域内の各位置情報（例えば、図１６に示す各位置情報）の設定などの前処理を実行する（Ｓ７０１）。
【００７３】
Ｓ７０１の処理を終えると、制御部１２１は、圧縮ファイルＦ２のコンテンツ部分を記憶領域Ｂ１にロードする（Ｓ７０２）。また、制御部１２１は、圧縮ファイルＦ２の終端に基づいて終了位置Ｑ２を設定する。次に、制御部１２１は、記憶領域Ｂ１の読出し位置Ｑ３の圧縮データに含まれる識別子が、最長一致データ列に基づく圧縮データでないこと（識別子「０」）を示すか、最長一致データ列に基づく圧縮データであること（識別子「１」）を示すかを判定する（Ｓ７０３）。
【００７４】
識別子が「０」である場合（Ｓ７０３：Ｙｅｓ）は、制御部１２１は、読出し位置Ｑ３の圧縮データに含まれている固定長符号を読出し、記憶領域Ｂ４に格納する（Ｓ７０４）。例えば、記憶領域Ｂ４に格納する固定長符号を固定長符号Ｍ（０）などとする。また、変換対象の固定長符号の数を示す移動量Ｌｃは、１とする（Ｌｃ＝１）。
【００７５】
識別子が「１」である場合（Ｓ７０３：Ｎｏ）は、制御部１２１は、参照部１２２に、読出し位置Ｑ３の圧縮データに含まれる位置Ｐａおよび長さＬａに基づき、記憶領域Ｂ２を参照させる。参照部１２２は、記憶領域Ｂ２の位置Ｐａから長さＬａの固定長符号列を読出し、記憶領域Ｂ４に格納する（Ｓ７０５）。記憶領域Ｂ４に格納される固定長符号列をＭ（０）～Ｍ（Ｌｃ－１）とする。Ｓ７０５において、制御部１２１は、移動量ＬｃをＬａに設定する（Ｌｃ＝Ｌａ）。
【００７６】
Ｓ７０４またはＳ７０５が行なわれると、制御部１２１は、変換部１２４に、記憶領域Ｂ４内に格納された固定長符号Ｍ（０）～Ｍ（Ｌｃ－１）のそれぞれについて、符号化辞書Ｄ１に基づく変換を実行させる（Ｓ７０６）。Ｓ７０４において、変換部１２４は、固定長符号の値に基づいて、符号化辞書Ｄ１内の位置を特定し、伸張データ（文字コード）を読み出す。図５の符号化辞書Ｄ１の例によれば、固定長符号の値が０ｘ０２０である場合は、「ａ」の文字コードが読み出される。

【００７７】
Ｓ７０６で伸張データが読み出されると、制御部１２１は、読み出された伸張データのそれぞれを記憶領域Ｂ３の書込み位置Ｑ１０に書き込む（Ｓ７０７）。さらに、制御部１２１は、書き込まれた伸張データの長さに応じて、書込み位置Ｑ１０を更新する。Ｓ７０７の処理が行なわれると、制御部１２１は、更新部１２３に記憶領域Ｂ２の更新を実行させる（Ｓ７０８）。
【００７８】
図１８は、記憶領域Ｂ２の更新処理の手順例を示す。更新部１２３は、制御部１２１に記憶領域Ｂ２の更新処理を指示される（Ｓ８００）と、カウンタ値ｉを初期値（ｉ＝０）にセットする（Ｓ８０１）。次に、更新部１２３は、記憶領域Ｂ２の更新位置Ｑ７からカウンタ値ｉに応じて移動された位置に、記憶領域Ｂ４に格納された固定長符号Ｍ（ｉ）を書き込む（Ｓ８０２）。具体的には、Ｓ８０２で書き込まれる位置は、固定長符号の符号長ｍとすると、更新位置Ｑ７からｍ×ｉビットずれた位置である。言い換えると、Ｓ８０２で書き込まれる位置は、更新位置Ｑ７を固定長符号の符号長ｍを単位として表現すると、Ｑ７＋ｉで表される位置である。
【００７９】
次に、更新部１２３は、カウンタ値ｉが移動量Ｌｃから１引いた値に達したか否かを判

定する（Ｓ８０３）。カウンタ値ｉが移動量Ｌｃから１引いた値になるまで処理が行なわれることによって、記憶領域Ｂ４に格納された固定長符号のそれぞれが記憶領域Ｂ２に反映される。

【００８０】

カウンタ値ｉが移動量Ｌｃから１引いた値に達していない場合（Ｓ８０３：Ｎｏ）に、更新部１２３はカウンタ値ｉをインクリメントする（Ｓ８０４）。さらに、更新部１２３は、Ｓ８０４でインクリメントされたカウンタ値ｉに基づいて、更新位置Ｑ７＋カウンタ値ｉが記憶領域Ｂ２の終了位置Ｑ５に達しているかを判断する（Ｓ８０５）。更新位置Ｑ７＋カウンタ値ｉが記憶領域Ｂ２の終了位置Ｑ５に達している場合（Ｓ８０５：Ｙｅｓ）は、更新部１２３は、更新位置Ｑ７に、記憶領域Ｂ２の開始位置Ｑ４からカウンタ値ｉを引いた値を代入する（Ｓ８０６）。Ｓ８０５およびＳ８０６の処理により、記憶領域Ｂ２外にはみ出して固定長符号が格納されることもなく、記憶領域Ｂ２が繰り返し使用される。更新位置Ｑ７＋カウンタ値ｉが記憶領域Ｂ２の終了位置Ｑ５に達していない場合（Ｓ８０５：Ｎｏ）か、Ｓ８０６の処理が行なわれた場合には、更新部１２３は、Ｓ８０２の処理を行なう。

【００８１】

カウンタ値ｉが移動量Ｌｃから１引いた値に達した場合（Ｓ８０３：Ｙｅｓ）に、更新部１２３は、記憶領域Ｂ２の更新位置Ｑ７を更新する（Ｓ８０７）。具体的には、更新位置Ｑ７に、更新位置Ｑ７に移動量Ｌｃを加算した値が代入される。Ｓ８０７の処理を終えると、更新部１２３は、記憶領域Ｂ２の更新処理を終了する（Ｓ８０８）。Ｓ８０８において、更新部１２３は、記憶領域Ｂ４内の情報をクリアする。

【００８２】

更新部１２３による記憶領域Ｂ２の更新処理が終了すると、制御部１２１は、伸張処理が圧縮ファイルＦ２の終点に達しているか判断する（Ｓ７０９）。Ｓ７０９は、例えば、記憶領域Ｂ１の読出し位置Ｑ３が記憶領域Ｂ１の終了位置Ｑ２に達しているか否かに応じて判断される。読出し位置Ｑ３が終了位置Ｑ２に達していない場合（Ｓ７０９：Ｎｏ）には、制御部１２１は、Ｓ７０３の処理を実行する。読出し位置Ｑ３が終了位置Ｑ２に達した場合（Ｓ７０９：Ｙｅｓ）には、制御部１２１は、記憶領域Ｂ３に格納された伸張データを用いて伸張ファイルＦ３を生成し、記憶部１３に格納する（Ｓ７１０）。すなわち伸張ファイルＦ３をクローズする。Ｓ７１０の処理が終了すると、制御部１２１は、伸張処理を終了する（Ｓ７１１）。Ｓ７１１の処理において、制御部１２１は、例えば、伸張機能の呼び出し元に対して、伸張処理の終了通知を行なう。伸張処理の終了通知には、例えば、伸張ファイルＦ３の格納先を示す情報などが含まれる。

【００８３】

下記に、本実施形態に用いられるハードウェア及びソフトウェアについて説明する。

【００８４】

図１９は、コンピュータ１のハードウェア構成例を示す。コンピュータ１は、例えば、プロセッサ３０１、ＲＡＭ（Ｒａｎｄｏｍ　Ａｃｃｅｓｓ　Ｍｅｍｏｒｙ）３０２、ＲＯＭ（Ｒｅａｄ　Ｏｎｌｙ　Ｍｅｍｏｒｙ）３０３、ドライブ装置３０４、記憶媒体３０５、入力インターフェース（Ｉ／Ｆ）３０６、入力デバイス３０７、出力インターフェース（Ｉ／Ｆ）３０８、出力デバイス３０９、通信インターフェース（Ｉ／Ｆ）３１０、ＳＡＮ（Ｓｔｏｒａｇｅ　Ａｒｅａ　Ｎｅｔｗｏｒｋ）インターフェース（Ｉ／Ｆ）３１１およびバス３１２などを含む。それぞれのハードウェアはバス３１２を介して接続されている。

【００８５】

ＲＡＭ３０２は読み書き可能なメモリ装置であって、例えば、ＳＲＡＭ（Ｓｔａｔｉｃ　ＲＡＭ）やＤＲＡＭ（Ｄｙｎａｍｉｃ　ＲＡＭ）などの半導体メモリ、またはＲＡＭでなくてもフラッシュメモリなどが用いられる。ＲＯＭ３０３は、ＰＲＯＭ（Ｐｒｏｇｒａｍｍａｂｌｅ　ＲＯＭ）なども含む。ドライブ装置３０４は、記憶媒体３０５に記録された情報の読み出しか書き込みかの少なくともいずれか一方を行なう装置である。記憶媒体

３０５は、ドライブ装置３０４によって書き込まれた情報を記憶する。記憶媒体３０５は、例えば、ハードディスク、ＳＳＤ（Ｓｏｌｉｄ Ｓｔａｔｅ Ｄｒｉｖｅ）などのフラッシュメモリ、ＣＤ（Ｃｏｍｐａｃｔ Ｄｉｓｃ）、ＤＶＤ（Ｄｉｇｉｔａｌ Ｖｅｒｓａｔｉｌｅ Ｄｉｓｃ）、ブルーレイディスクなどの記憶媒体である。また、例えば、コンピュータ１は、複数種類の記憶媒体それぞれについて、ドライブ装置３０４及び記憶媒体３０５を設ける。

【００８６】

入力インターフェース３０６は、入力デバイス３０７と接続されており、入力デバイス３０７から受信した入力信号をプロセッサ３０１に伝達する回路である。出力インターフェース３０８は、出力デバイス３０９と接続されており、出力デバイス３０９に、プロセッサ３０１の指示に応じた出力を実行させる回路である。通信インターフェース３１０はネットワーク３を介した通信の制御を行なう回路である。通信インターフェース３１０は、例えばネットワークインターフェースカード（ＮＩＣ）などである。ＳＡＮインターフェース３１１は、ストレージエリアネットワークによりコンピュータ１と接続された記憶装置との通信の制御を行なう回路である。ＳＡＮインターフェース３１１は、例えばホストバスアダプタ（ＨＢＡ）などである。

【００８７】

入力デバイス３０７は、操作に応じて入力信号を送信する装置である。入力デバイス３０７は、例えば、キーボードやコンピュータ１の本体に取り付けられたボタンなどのキー装置や、マウスやタッチパネルなどのポインティングデバイスである。出力デバイス３０９は、コンピュータ１の制御に応じて情報を出力する装置である。出力デバイス３０９は、例えば、ディスプレイなどの画像出力装置（表示デバイス）や、スピーカーなどの音声出力装置などである。また、例えば、タッチスクリーンなどの入出力装置が、入力デバイス３０７及び出力デバイス３０９として用いられる。また、入力デバイス３０７及び出力デバイス３０９は、コンピュータ１と一体になっていてもよいし、コンピュータ１に含まれず、例えば、コンピュータ１に外部から接続する装置であってもよい。

【００８８】

例えば、プロセッサ３０１は、ＲＯＭ３０３や記憶媒体３０５に記憶されたプログラムをＲＡＭ３０２に読み出し、読み出されたプログラムの手順に従って圧縮部１１の処理または伸張部１２の処理を行なう。その際にＲＡＭ３０２はプロセッサ３０１のワークエリアとして用いられる。記憶部１３の機能は、ＲＯＭ３０３および記憶媒体３０５がプログラムファイル（後述のアプリケーションプログラム２４、ミドルウェア２３およびＯＳ２２など）やデータファイル（圧縮対象のファイルＦ１、圧縮ファイルＦ２および伸張ファイルＦ３など）を記憶し、ＲＡＭ３０２がプロセッサ３０１のワークエリアとして用いられることによって実現される。プロセッサ３０１が読み出すプログラムについては、図２０を用いて後述する。

【００８９】

図１０～図１５の処理を実行する圧縮部１１内の各機能ブロックについて、さらに説明する。制御部１１１は、プロセッサ３０１がＲＡＭ３０２の制御（排他制御など）や、ＲＡＭ３０２へのアクセス処理や、アクセス処理で得られた情報に対する演算や、プロセッサ３０１内での演算処理などを行なうことにより実現される。照合部１１２は、プロセッサ３０１がＲＡＭ３０２へのアクセス処理や、アクセス処理により得られた情報に対する照合の演算などを行なうことにより実現される。更新部１１３は、プロセッサ３０１によるＲＡＭ３０２へのアクセス処理などを行なうことにより実現される。変換部１１４は、プロセッサ３０１によるＲＡＭ３０２へのアクセス処理や、アクセス処理により得られた情報に対する照合の演算等を行なうことにより実現される。

【００９０】

図１７および図１８の処理を実行する伸張部１２内の各機能ブロックについて、さらに説明する。制御部１２１は、プロセッサ３０１がＲＡＭ３０２の制御（排他制御など）や

、ＲＡＭ３０２へのアクセス処理や、アクセス処理で得られた情報に対する演算や、プロセッサ３０１内での演算処理などを行なうことにより実現される。参照部１２２は、プロセッサ３０１がＲＡＭ３０２へのアクセス処理などを行なうことにより実現される。更新部１２３は、プロセッサ３０１によるＲＡＭ３０２へのアクセス処理などを行なうことにより実現される。変換部１２４は、プロセッサ３０１によるＲＡＭ３０２へのアクセス処理や、アクセス処理により得られた情報に対する照合の演算等を行なうことにより実現される。

【００９１】

図２０は、コンピュータ１で動作するプログラムの構成例を示す。コンピュータ１において、図１９に示すハードウェア群２１（３０１～３１２）の制御を行なうＯＳ（オペレーティング・システム）２２が動作する。ＯＳ２２に従った手順でプロセッサ３０１が動作して、ハードウェア群２１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２４やミドルウェア２３に従った処理がハードウェア群２１で実行される。さらに、コンピュータ１において、ミドルウェア２３またはアプリケーションプログラム２４が、ＲＡＭ３０２に読み出されてプロセッサ３０１により実行される。

【００９２】

プロセッサ３０１が、圧縮機能が呼び出された場合に、ミドルウェア２３またはアプリケーションプログラム２４の少なくとも一部に基づく処理を行なうことにより、（それらの処理をＯＳ２２に基づいてハードウェア群２１を制御して）圧縮部１１の機能が実現される。また、プロセッサ３０１が、伸張機能が呼び出された場合に、ミドルウェア２３またはアプリケーションプログラム２４の少なくとも一部に基づく処理を行なうことにより、（それらの処理をＯＳ２２に基づいてハードウェア群２１を制御して）伸張部１２の機能が実現される。圧縮機能および伸張機能は、それぞれアプリケーションプログラム２４自体に含まれてもよいし、アプリケーションプログラム２４に従って呼び出されることで実行されるミドルウェア２３の一部であってもよい。もしくは圧縮機能および伸張機能がＯＳ２２の一機能であってもよい。

【００９３】

アプリケーションプログラム２４（またはミドルウェア２３）の圧縮機能では、処理対象のデータに合致するデータを抽出するための照合回数が抑制されるため、プロセッサ３０１のメモリアクセスの負荷が抑制される。そのため、ＲＡＭ３０２上にワークエリアを確保する時間も削減される。

【００９４】

図２１は、実施形態のシステムにおける装置の構成例を示す。図２１のシステムは、コンピュータ１ａ、コンピュータ１ｂ、基地局２およびネットワーク３を含む。コンピュータ１ａ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１ｂと接続されたネットワーク３に接続している。

【００９５】

図８に示す圧縮部１１と伸張部１２とは、図２１に示すコンピュータ１ａとコンピュータ１ｂとのいずれに含まれてもよい。例えば、コンピュータ１ｂが圧縮部１１（制御部１１１、照合部１１２、更新部１１３および変換部１１４を含む）を含み、コンピュータ１ａが伸張部１２（制御部１２１、参照部１２２、更新部１２３および変換部１２４を含む）を含んでもよいし、コンピュータ１ａが圧縮部１１を含み、コンピュータ１ｂが伸張部１２を含んでもよい。また、コンピュータ１ａとコンピュータ１ｂとの双方が、圧縮部１１および伸張部１２を備えてもよい。

【００９６】

文字コードにおける位置が異なるデータ間の照合が発生する例について、図２２および図２３に基づいて補足説明する。

【００９７】

ＵＴＦ－８のコード割り当てでは、２バイト以上の文字コードにおいて、２バイト目以

降のバイトでは、値の範囲が共通している（いずれも０ｘ８０～０ｘＢＦの範囲内である）。そのため、複数バイトで文字を表現する文字コードを用いたデータ同士で、バイト単位で照合を行なうと、異なる文字コードであっても一部分だけ一致することがあり得る。例えば、ある４バイト文字コードの３番目のバイトと、他の３バイト文字の２番目のバイトとが一致するなどの事態が発生する。すると、図２２および図２３に例示されるような照合処理が発生しうる。

【００９８】

図２２は、圧縮対象のデータを構成するデータ単位と異なるデータ単位での照合処理例を示す。図２２は、記憶領域Ａ１及び記憶領域Ａ２のそれぞれを部分的に示す。各記憶領域内の点線での区切りは１バイト単位での区切りを示し、実線での区切りは文字コードの区切りを示す。図２２の例においては、各記憶領域内のデータとして３バイトの文字コードが例示されている。

【００９９】

例えば、図２２に示すように、処理対象のデータを読み出す記憶領域Ａ１内の位置が読出し位置Ｐ３（１）であるとし、処理対象のデータと照合される記憶領域Ａ２内のデータの位置が参照位置Ｐ６（１）であるとする。図２２に例示するように、３バイトの文字コード間の照合を１バイト単位で行なうと、最長一致データの終端が文字コードの区切りとは異なる位置に存在することがあり得る。図２２には、３バイトの文字コードが２文字分と、３バイトの文字コード内の２バイト分が最長一致データとして抽出された場合が例示されている。ＬＺ７７を利用した圧縮処理においては、抽出された最長一致データの位置と長さに基づいて圧縮符号が生成されるので、参照位置Ｐ６（１）と最長一致データの長さ（８バイト）とに基づいて圧縮符号が生成される。

【０１００】

図２２に示す最長一致データに基づいて圧縮符号が生成されると、処理対象を読み出す記憶領域Ａ１内の位置が、読出し位置Ｐ３（１）から読出し位置Ｐ３（２）に更新される。続いて、読出し位置Ｐ３（２）からのデータに基づいて、最長一致データの探索が行なわれる。

【０１０１】

図２３は、圧縮対象のデータを構成するデータ単位と異なるデータ単位での照合処理例を示す。図２３は、記憶領域Ａ１及び記憶領域Ａ２のそれぞれを部分的に示す。読出し位置Ｐ３（２）のデータは、「１０ＸＸＸＸＸＸ」であり、ＵＴＦ－８の文字コード系においては、２バイト目以降のデータである。例えば、読出し位置Ｐ３（２）のデータ（「１０ＸＸＸＸＸＸ」）と合致する記憶領域Ａ２内のデータが、図２３に示すように、参照位置Ｐ６（２１）や参照位置Ｐ６（２２）に存在したとする。図２３の例においては、参照位置Ｐ６（２１）のデータは、３バイトの文字コードの３バイト目のデータであり、参照位置Ｐ６（２２）のデータは、３バイトの文字コードの２バイト目のデータである。

【０１０２】

読出し位置Ｐ３（２）のデータと参照位置Ｐ６（２１）のデータとの一致に応じて、読出し位置Ｐ３（２）のデータに後続するデータ（図２３の例では「１１１０ＹＹＹＹ」）と、参照位置Ｐ６（２１）のデータに後続するデータ（図２３の例では「１１１０ＹＹＹＹ」）との照合が行なわれる。この照合では、両方のデータが３バイトの文字コードの１バイト目であるため、照合により一致する可能性がある。

【０１０３】

読出し位置Ｐ３（２）のデータと参照位置Ｐ６（２２）のデータとの一致に応じて、読出し位置Ｐ３（２）のデータに後続するデータ（図２３の例では「１１１０ＹＹＹＹ」）と、参照位置Ｐ６（２２）のデータに後続するデータ（図２３の例では「１０ＸＸＸＸＸＸ」）との照合が行なわれる。この照合では、一方が３バイトの文字コードの１バイト目であり、他方が３バイトの文字コードの３バイト目であるため、照合により一致しないことが明らかである。

【０１０４】
図２２および図２３に示される例においては、３バイトの文字コード間の照合を１バイト単位で行なうことにより、文字コードの区切りとは異なる位置で最長一致データが区切られる。すると、図２３に示すように、文字コードにおける位置が異なるデータ間の照合が発生する可能性がある。しかしながら、例えば、３バイトの文字コードの１バイト目のデータと３バイト目のデータとは、文字コードの体系の都合上明らかに一致しないにも関わらず、照合処理が行なわれてしまう。
【０１０５】
一方、上述の実施形態によれば、照合処理の単位が文字コード単位で行なわれるため、明らかに異なるデータ同士の照合処理が行なわれてしまうことが抑止される。
【０１０６】
以下、上述の実施形態における変形例の一例を説明する。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なわれうる。
【０１０７】
図２４は、Ｓ３０１からＳ３０３の処理の例を示す。変換部１１４は、図１２に示すＳ３０１～Ｓ３０３の処理を、ファイルＦ１に用いられる文字コードが例えばＵＴＦ－８である場合に、以下の手順で実行する。
【０１０８】
Ｓ３００が行なわれる（Ｓ９００）と、変換部１１４は、記憶領域Ａ１の読出し位置Ｐ３から１バイトのデータを読み出す（Ｓ９０１）。変換部１１４は、読みだしたデータの１ビット目が「１」であるか否かを判定する（Ｓ９０２）。Ｓ９０１で読みだしたデータの１ビット目が「１」でない（「０」である）場合（Ｓ９０２：Ｎｏ）には、変換部１１４は、移動量Ｌｄに１を代入する（Ｓ９０３）。移動量Ｌｄは、後述の読出し位置Ｐ３の更新に用いられる。
【０１０９】
Ｓ９０１で読みだしたデータの１ビット目が「１」である場合（Ｓ９０２：Ｙｅｓ）には、変換部１１４は、読みだしたデータの３ビット目が「１」であるか否かを判定する（Ｓ９０４）。Ｓ９０１で読みだしたデータの３ビット目が「１」でない（「０」である）場合（Ｓ９０４：Ｎｏ）には、変換部１１４は、移動量Ｌｄに２を代入し、記憶領域Ａ１からさらに１バイトのデータを読み出す（Ｓ９０５）。
【０１１０】
Ｓ９０１で読みだしたデータの３ビット目が「１」である場合（Ｓ９０４：Ｙｅｓ）には、変換部１１４は、読みだしたデータの４ビット目が「１」であるか否かを判定する（Ｓ９０６）。Ｓ９０１で読みだしたデータの４ビット目が「１」でない（「０」である）場合（Ｓ９０６：Ｎｏ）には、変換部１１４は、移動量Ｌｄに３を代入し、記憶領域Ａ１からさらに２バイトのデータを読み出す（Ｓ９０７）。
【０１１１】
Ｓ９０１で読みだしたデータの４ビット目が「１」である場合（Ｓ９０６：Ｙｅｓ）には、変換部１１４は、移動量Ｌｄに４を代入し、記憶領域Ａ１からさらに３バイトのデータを読み出す（Ｓ９０８）。
【０１１２】
Ｓ９０３、Ｓ９０５、Ｓ９０７およびＳ９０８のいずれかが行なわれると、変換部１１４は、移動量Ｌｄに基づいてインデックスＥ１を参照し、参照結果を利用して符号化辞書Ｄ１から、読みだしたデータに対応する固定長符号を読み出す（Ｓ９０９）。インデックスＥ１については、図２５を用いて後述する。変換部１１４は、さらに、読出し位置Ｐ３を移動量Ｌｄに示される量（Ｌｄバイト）移動させる（Ｓ９１０）。Ｓ９１０の処理を終えると変換部１１４はＳ３０４の処理を実行する。
【０１１３】
図２５は、符号化辞書Ｄ１のインデックス例を示す。図２５に示すインデックスＥ１は、移動量Ｌｄが１～４の場合それぞれについて、符号化辞書Ｄ１内のサーチ開始位置を示

す。例えば、移動量Ｌｄが１の場合には、変換部１１４は、固定長符号０ｘ０００の位置から符号化辞書Ｄ１のサーチを開始する。移動量Ｌｄが２の場合には、変換部１１４は、固定長符号０ｘ１００の位置から符号化辞書Ｄ１のサーチを開始する。移動量Ｌｄが３の場合には、変換部１１４は、固定長符号０ｘ１８０の位置から符号化辞書Ｄ１のサーチを開始する。移動量Ｌｄが４の場合には、変換部１１４は、固定長符号０ｘ８００の位置から符号化辞書Ｄ１のサーチを開始する。符号化辞書Ｄ１に含まれる文字コードの長さの分布に応じてインデックスＥ１の値が設定されることで、異なる長さの文字コード同士の照合が抑制される。符号化辞書Ｄ２に対して図２５に示すインデックスと同様なインデックスを利用したサーチが行なわれてもよい。

【０１１４】

図２６は、最長一致固定長符号列の探索処理の変形例を示す。図２６の変形例においては、記憶領域Ａ２内の各固定長符号に対応したビットを含むビット列Ｒ１～Ｒ３が用いられる。ビット列Ｒ１～Ｒ３の記憶領域は、記憶部１３内に設けられる。記憶領域Ａ２内の各固定長符号に対して１ビット用いられるため、各ビット列のサイズは、記憶領域Ａ２の１／ｍである。

【０１１５】

ビット列Ｒ１は、照合対象の固定長符号Ｍ（ｊ）が記憶領域Ａ２内に含まれているか否かを示すビット列である。固定長符号Ｍ（ｊ）とは、前述の通り、記憶領域Ａ４内のｊ番目に格納された固定長符号である。すなわち、記憶領域Ａ２内の位置Ｐｘに固定長符号Ｍ（ｊ）と同じ固定長符号が格納されている場合には、ビット列Ｒ１のＰｘ番目のビットが「存在」を示す（値が「１」となる）。

【０１１６】

ビット列Ｒ２は、固定長符号Ｍ（０）～Ｍ（ｊ－１）までの照合結果を示すビット列である。また、ビット列Ｒ３は、ビット列Ｒ１とビット列Ｒ２との演算結果をしめす。具体的には、ビット列Ｒ１をｊビットスライド（図２６中の矢印の方向）させて、スライドしたビット列Ｒ１とビット列Ｒ２とのＡＮＤ演算の結果がビット列Ｒ３となる。ＡＮＤ演算が行なわれた後、ｊ＋１番目の処理のために、ビット列Ｒ３は、ビット列Ｒ２にコピーされる。具体的な手順は図２７を用いて説明するが、ビット列Ｒ１～Ｒ３を用いた上述の処理の繰り返すことで、「存在」を示すビットが最後まで残った位置により最長一致位置Ｐａが示される。さらに、繰り返し回数が一致長Ｌａを示す。

【０１１７】

図２７は、最長一致符号列の探索処理の手順例を示す。最長一致符号列の探索処理が開始される（Ｓ１０００）と、制御部１１１は、ビット列Ｒ１～Ｒ３を初期化する（Ｓ１００１）。さらに、制御部１１１は、一致長Ｌａおよび最長一致位置Ｐａに初期値（Ｌａ＝０，Ｐａ＝Ｐ４－１など）をセットする（Ｓ１００２）。さらに、制御部１１１は、カウンタ値ｊの初期値（ｊ＝０）をセットする（Ｓ１００３）。

【０１１８】

続いて、制御部１１１は、記憶領域Ａ４に固定長符号Ｍ（ｊ）が格納されているか否かを判断する（Ｓ１００４）。記憶領域Ａ４に固定長符号Ｍ（ｊ）が格納されていない場合（Ｓ１００４：Ｎｏ）には、制御部１１１は、固定長符号Ｍ（ｊ）の取得処理を変換部１１４に実行させる（Ｓ１００５）。変換部１１４は、図１２に記載の処理を実行する。

【０１１９】

記憶領域Ａ４に固定長符号Ｍ（ｊ）が格納されている場合（Ｓ１００４：Ｙｅｓ）か、Ｓ１００５の処理が実行された場合には、制御部１１１は、記憶領域Ａ２内における固定長符号Ｍ（ｊ）の存否結果をビット列Ｒ１に反映させる（Ｓ１００６）。例えば、制御部１１１は、記憶領域Ａ２中の固定長符号Ｍ（ｊ）と同じ固定長符号の存在位置に対応するビットを「１」に変更する。さらに、制御部１１１は、ビット列Ｒ１をｊビットスライドさせた（Ｓ１００７）後、ビット列Ｒ２とビット列Ｒ１とで、ビット列内の各ビットについてのＡＮＤ演算を行ない、その結果をビット列Ｒ３とする（Ｓ１００８）。

【０１２０】
続いて、制御部１１１は、ビット列Ｒ３内に存在（「１」）を示すビットが存在するか否かを判定する（Ｓ１００９）。ビット列Ｒ３内に存在（「１」）を示すビットが存在する場合（Ｓ１００９：Ｙｅｓ）には、制御部１１１は、ビット列Ｒ１をビット列Ｒ２にコピーし（Ｓ１０１０）、カウンタ値ｊをインクリメントし（Ｓ１０１１）、Ｓ１００４の処理を実行する。
【０１２１】
ビット列Ｒ３内に存在（「１」）を示すビットが存在しない場合（Ｓ１００９：Ｎｏ）には、制御部１１１は、ビット列Ｒ２内の存在（「１」）を示すビットのうち、いずれかの位置（何ビット目かを示す値）を最長一致位置Ｐａ（固定長符号が何個分かを示す値）に代入する（Ｓ１０１２）。さらに、制御部１１１は、一致長Ｌａにカウンタ値ｊを代入する（Ｓ１０１３）。Ｓ１０１３の処理が行なわれると、制御部１１１は、最長一致符号列の探索処理を終了する（Ｓ１０１４）。

【０１２２】
さらに、文字コード長と照合処理の単位との不一致による不要な照合処理が発生することを抑制する実施形態の変形例を説明する。例えば、ＵＴＦ－８においては、文字コードにおける始めの１バイトのデータにより文字コード長が判別される。例えば、図１０のＳ１０４の処理で、照合部１１２が、記憶領域Ａ１の読出し位置Ｐ３と記憶領域Ａ２の参照位置Ｐ６との双方における１バイトのデータに基づいて文字コード長の合致判定を行ない、合致すると判定された場合に文字コード単位での照合を行なってもよい。文字コード内の始めの１バイトにより文字コード長が判別されるため、照合部１１２は、合致すると判定されてから、記憶領域Ａ１の読出し位置Ｐ３と記憶領域Ａ２の参照位置Ｐ６との双方における文字コードを読出し、文字コード単位での照合を行なう。
【０１２３】
記憶領域Ａ１の読出し位置Ｐ３の文字コードと、記憶領域Ａ２の参照位置Ｐ６の文字コードとの双方で、文字コード長が合わない場合には、照合処理をスキップして参照位置Ｐ６の更新が行なわれる。参照位置Ｐ６の更新における参照位置Ｐ６の移動量は、例えば参照位置Ｐ６の文字コードの文字コード長である。
【０１２４】
また、この変形例の前提として、記憶領域Ａ２には、文字コードが格納される。すなわち、図１２のＳ３０４の処理において文字コードが記憶領域Ａ４に書き込まれ、さらに、図１４のＳ５０２において記憶領域Ａ４内の文字コードが記憶領域Ａ２に書き込まれる。さらに、例えば、読出し位置Ｐ３の移動量Ｌｄには、判別された文字コードのバイト数が用いられる。
【０１２５】
上述の通り、記憶領域Ａ１の読出し位置Ｐ３から読み出した文字コードと記憶領域Ａ２の参照位置Ｐ６から読みだした文字コードのバイト数が合わない場合に文字コード同士の照合をスキップすることで、不要な文字コード同士の照合が回避される。この変形例を用いる場合には、上述の通り、図１０のＳ１０６の処理では、記憶領域Ａ１から読み出された文字コードが記憶領域Ａ２に格納される。図１７のＳ７０８においては、固定長符号ではなく、伸張データが記憶領域Ｂ２に書き込まれる。また、Ｓ７０６の処理がスキップされる。
【０１２６】
さらに、他の変形例として、照合部１１２が、照合処理の実行単位は例えば１バイトで行ない、１バイトデータの照合の前に、１バイト文字コード内の位置が同一のデータであるかの判定を行なうこととしてもよい。文字コードによっては、文字コードの表現に用いられる各バイトのデータは、文字コード長および文字コード内での位置に応じて複数種類に分類される。例えば、図３に示す通り、ＵＴＦ－８においては、１バイト文字は「０ＸＸＸＸＸＸＸ」、２バイト文字の１バイト目は「１１０ＹＹＹＹＸ」、３バイト文字の１

バイト目は「１１１０ＹＹＹＹ」、４バイト文字の１バイト目は「１１１１０ＹＹＹ」、２～４バイト文字の２バイト目以降は「１０ＸＸＸＸＸＸ」である。「Ｘ」は不特定のビットを便宜的に表す。すなわち、ＵＴＦ－８では、１バイトのデータにおける先頭から数ビットのデータに応じて５種類に分類される。種類が異なる１バイトデータ間で照合処理を行なったとしても合致しないことは明らかなので、例えば、照合部１１２は、１バイトデータ間で種類が異なれば照合処理をスキップする。これにより不要な照合処理が抑制される。また、各文字コードの先頭バイトの種類が一致しているので、結果的に文字コード長が合致したデータ間の照合処理により、最長一致データ列が抽出される。この変形例も記憶領域Ａ２内に文字コードが格納されることを前提としている。そのため、記憶領域Ａ２の更新処理について、先に説明した変形例と同様の制御が行なわれる。

【０１２７】

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。

【０１２８】

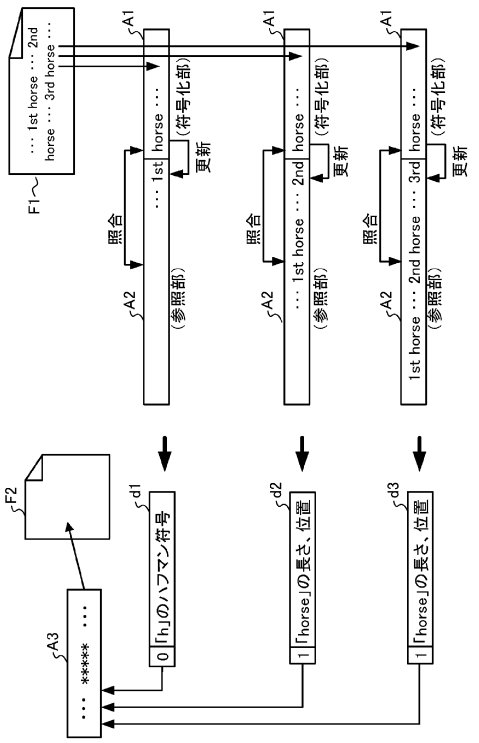
また、上述の圧縮処理の対象となるデータは、上述の通り、文字情報に限定されるものでない。数値のみの情報であってもよいし、画像・音声などのデータに対して上述の圧縮処理を用いてもよい。例えば、音声合成により得られるデータを多量に含むファイルなどは、データ内に繰り返しを多く含むため動的辞書により圧縮率が向上することが見込まれる。また、固定カメラにより撮影された動画像についても各フレームの画像が似たものになることから繰り返しが多く含まれる。そのため、上述の圧縮処理を適用することにより、文書データや音声データと同様の効果を得ることができる。

【符号の説明】

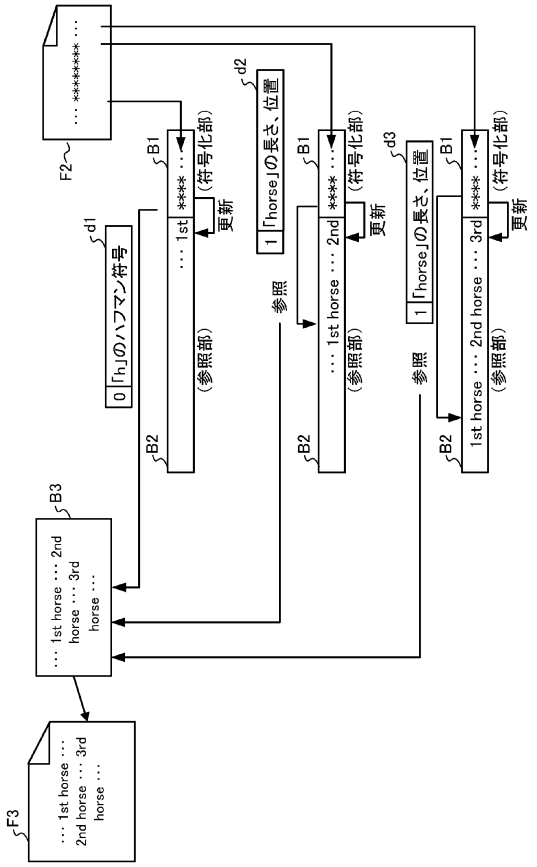
【０１２９】

１　コンピュータ
２　基地局
３　ネットワーク
１ａ　コンピュータ
１ｂ　コンピュータ
１１　圧縮部
１２　伸張部
１３　記憶部
１１１　制御部
１１２　照合部
１１３　更新部
１１４　変換部
１２１　制御部
１２２　参照部
１２３　更新部
１２４　変換部

【図１】



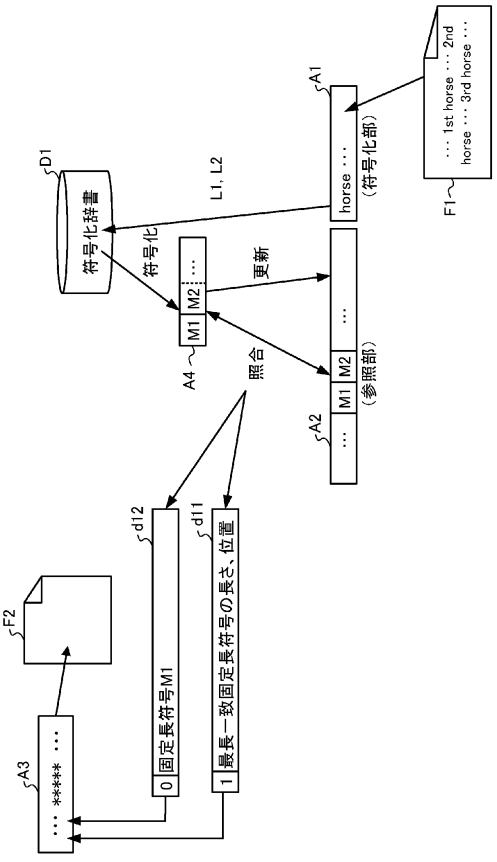
【図２】



【図３】

1バイト目	2バイト目	3バイト目	4バイト目	文字の種類
00-7F 0XXXXXXX				制御コード ASCII文字
C2-DF 110YYYYX	80-BF 10XXXXXX			UCS-2マルチバイト文字 (0x0080-0x07FF)
E0-EF 1110YYYY	80-BF 10YXXXXX	80-BF 10XXXXXX		UCS-2マルチバイト文字 (0x0800-0xFFFF)
F0-F7 11110YYY	80-BF 10YYXXXX	80-BF 10XXXXXX	80-BF 10XXXXXX	UCS-4マルチバイト文字 (0x00010000-0x0010FFFF)

【図４】



【図５】

D1

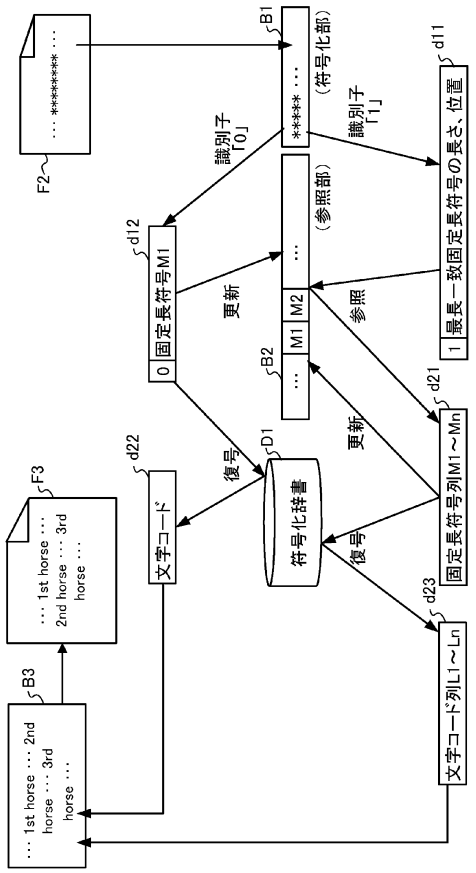
固定長符号（登録番号）	データ（文字コード）
0x000	0x00（“NUL”の文字コード）
⋮	⋮
0x010	0x30（“0”の文字コード）
0x011	0x31（“1”の文字コード）
⋮	⋮
0x020	0x41（“a”の文字コード）
0x021	0x42（“b”の文字コード）
⋮	⋮
0x100	0xCEB1（“α”の文字コード）
0x101	0xCEB2（“β”の文字コード）
⋮	⋮
0x180	0xE38182（“あ”の文字コード）
0x181	0xE38183（“い”の文字コード）
⋮	⋮
0x240	0xE4B880（“一”の文字コード）
0x241	0xE4B881（“丁”の文字コード）
⋮	⋮
0xFFE	0xE9BCBB（“鼻”の文字コード）
0xFFF	0xE9BE8D（“龍”の文字コード）

【図６】

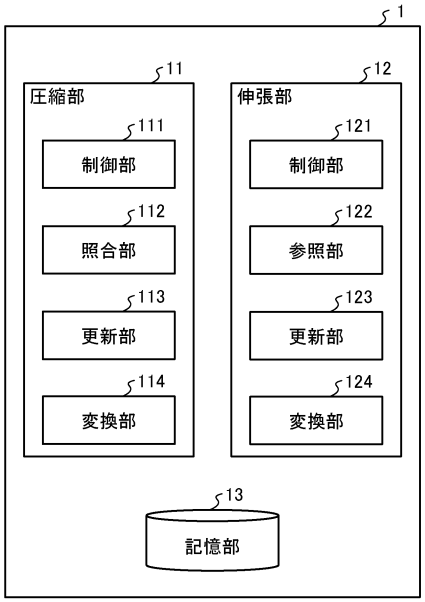
D2

固定長符号（登録番号）	データ（文字コード）
0x000	0x00（“NUL”の文字コード）
⋮	⋮
0x010	0x30（“0”の文字コード）
0x011	0x31（“1”の文字コード）
⋮	⋮
0x020	0x41（“a”の文字コード）
0x021	0x42（“b”の文字コード）
⋮	⋮
0x100	0x6F6E65（“one”の文字コード）
0x101	0x74776F（“two”の文字コード）
⋮	⋮
0x180	0x617265（“are”の文字コード）
0x181	0x77657265（“were”の文字コード）
⋮	⋮
0x200	0x74686973（“this”の文字コード）
0x201	0x74686174（“that”の文字コード）
⋮	⋮
0x280	0x77616B65（“wake”の文字コード）
0x281	0x736C656570（“sleep”の文字コード）
⋮	⋮
0x300	0x6D6F726E696E67（“morning”の文字コード）
0x301	0x6E69676874（“night”の文字コード）
⋮	⋮
0xFFE	0x7A6F6E65（“zone”の文字コード）
0xFFF	0x7A6F6F（“zoo”の文字コード）

【図７】



【図８】

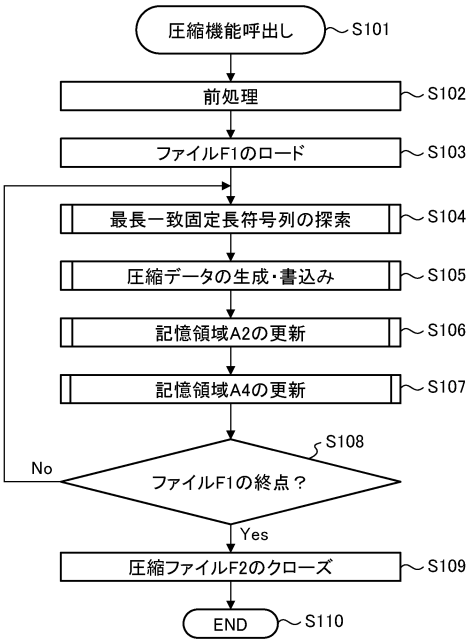


【図９】

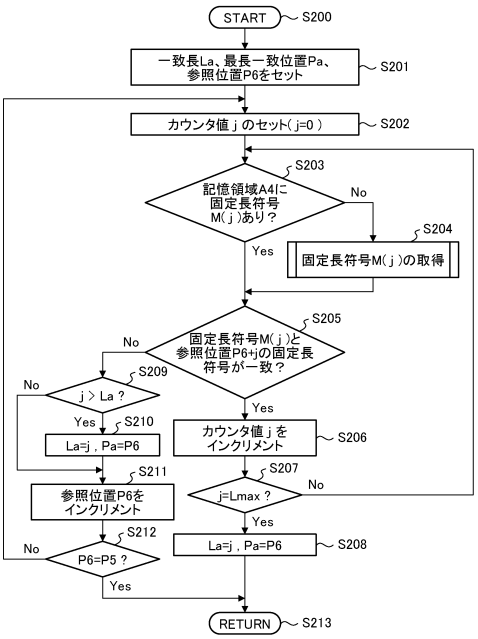
T1

記憶領域A1（ファイルF1）	開始位置	P1
	終了位置	P2
	読出し位置	P3
記憶領域A2	開始位置	P4
	終了位置	P5
	参照位置	P6
	更新位置	P7
記憶領域A3（ファイルF2）	開始位置	P8
	終了位置	P9
	書込み位置	P10

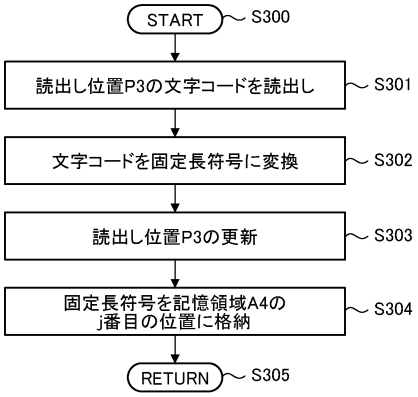
【図１０】



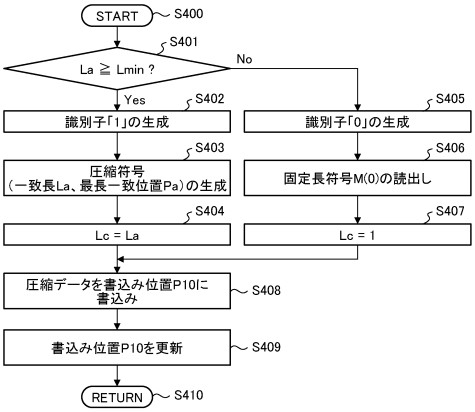
【図１１】



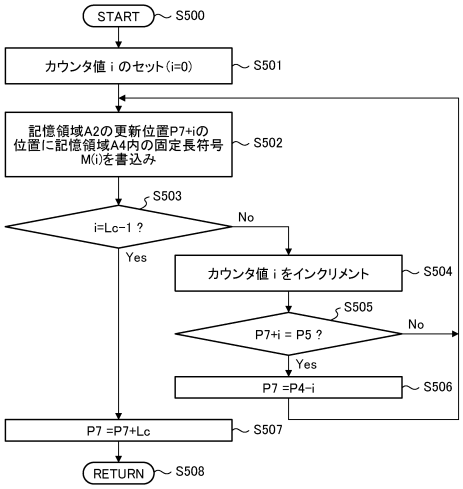
【図１２】



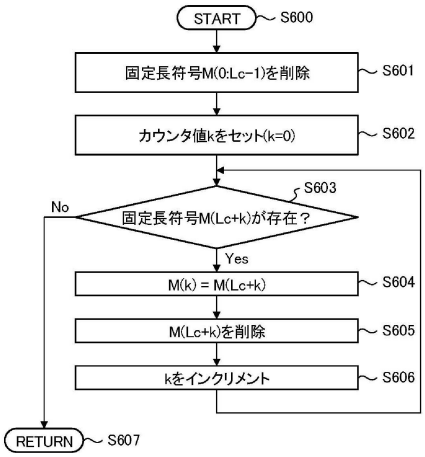
【図１３】



【図１４】



【図１５】

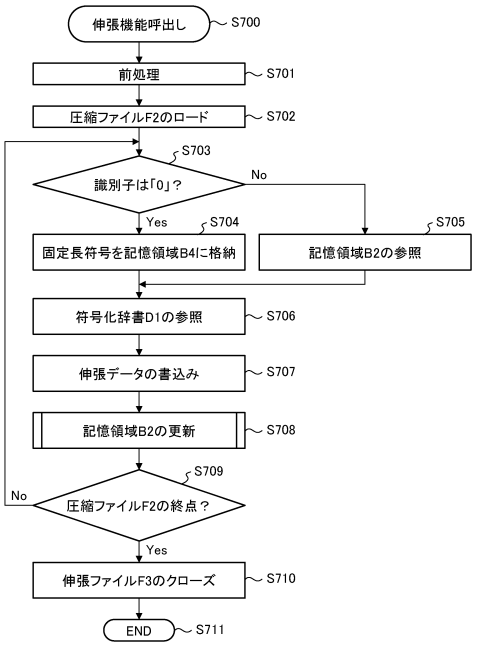


【図１６】

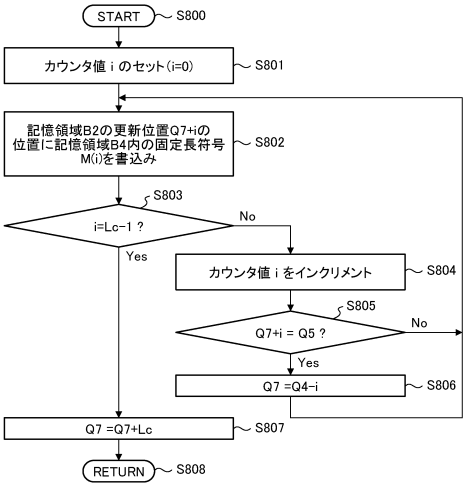
T2

記憶領域B1 （ファイルF2）	開始位置	Q1
	終了位置	Q2
	読出し位置	Q3
記憶領域B2	開始位置	Q4
	終了位置	Q5
	参照位置	Q6
	更新位置	Q7
記憶領域B3 （ファイルF3）	開始位置	Q8
	終了位置	Q9
	書込み位置	Q10

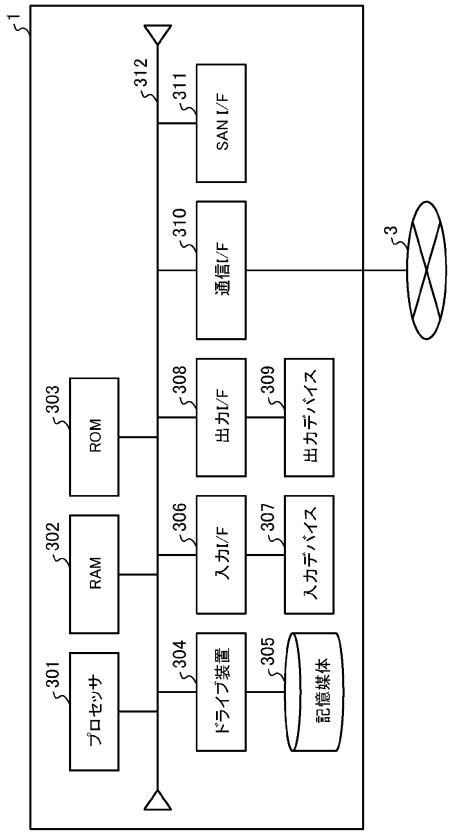
【図１７】



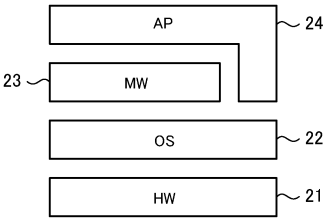
【図１８】



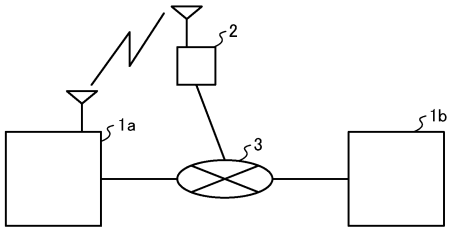
【図１９】



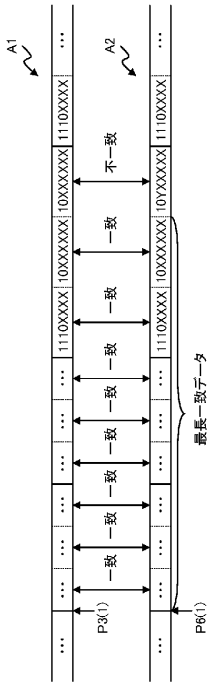
【図２０】



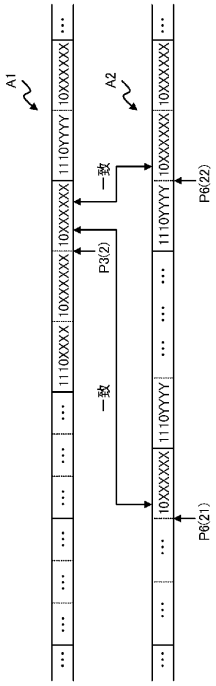
【図２１】



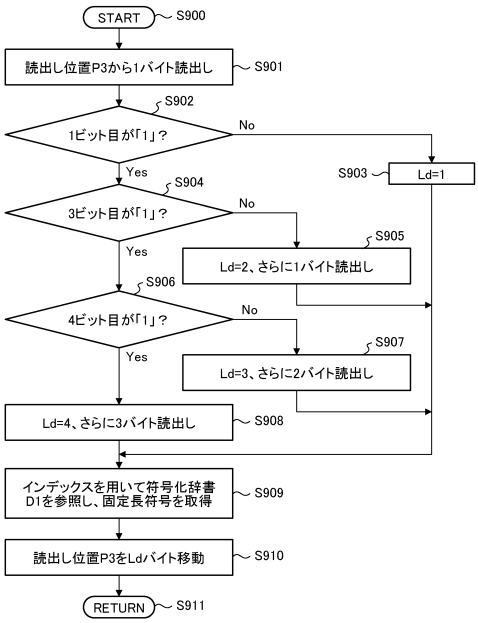
【図２２】



【図２３】



【図２４】

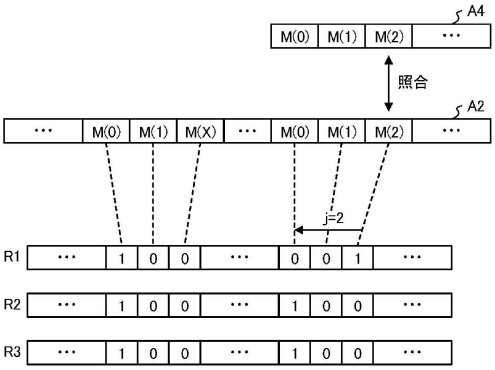


【図２５】

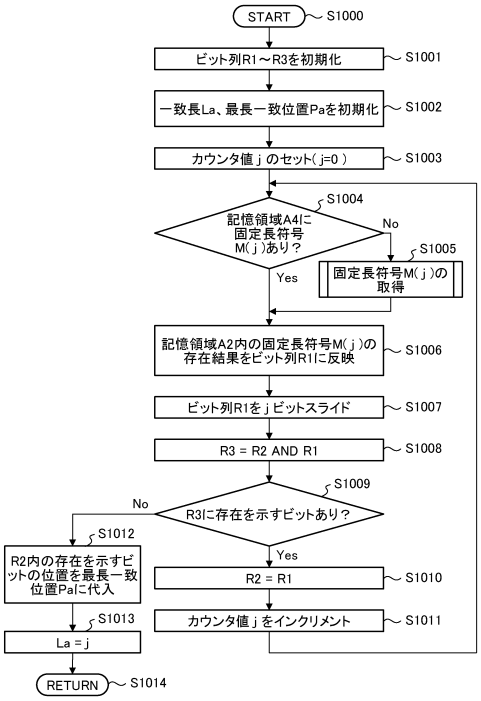
E1

移動量 Ld	固定長符号（登録番号）
1	0x000
2	0x100
3	0x180
4	0x800

【図２６】


A4
M(0)
M(1)
M(2)
照合
A2
M(0)
M(1)
M(X)
M(0)
M(1)
M(2)
j=2
R1
R2
R3

【図２７】


START
S1000
ビット列R1～R3を初期化
S1001
一致長La、最長一致位置Paを初期化
S1002
カウンタ値ｊのセット(j=0)
S1003
S1004
記憶領域A4に固定長符号M(j)あり？
No
Yes
S1005
固定長符号M(j)の取得
記憶領域A2内の固定長符号M(j)の存在結果をビット列R1に反映
S1006
ビット列R1をｊビットスライド
S1007
R3 = R2 AND R1
S1008
S1009
R3に存在を示すビットあり？
No
Yes
S1012
R2内の存在を示すビットの位置を最長一致位置Paに代入
S1013
La = j
R2 = R1
S1010
カウンタ値ｊをインクリメント
S1011
RETURN
S1014

フロントページの続き

審査官　桜井　茂行

(56)参考文献　特開平１１－２１５００７（ＪＰ，Ａ）
米国特許第６５４２６４０（ＵＳ，Ｂ１）
特開平０９－２１８８６７（ＪＰ，Ａ）
欧州特許出願公開第０７８９４６０（ＥＰ，Ａ１）
特開２００２－１３５１２８（ＪＰ，Ａ）
特開平６－１６８０９７（ＪＰ，Ａ）
特開平５－２４１７７６（ＪＰ，Ａ）
特開平９－２１４３５２（ＪＰ，Ａ）
欧州特許出願公開第０７８８２３９（ＥＰ，Ａ２）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１２／００
Ｇ０６Ｆ　５／００