

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4000266号
(P4000266)

(45) 発行日 平成19年10月31日(2007.10.31)

(24) 登録日 平成19年8月17日(2007.8.17)

(51) Int.C1.

F 1

HO4N 1/41 (2006.01)
HO3M 7/42 (2006.01)HO4N 1/41
HO3M 7/42

B

請求項の数 9 (全 13 頁)

(21) 出願番号 特願2002-64271 (P2002-64271)
 (22) 出願日 平成14年3月8日 (2002.3.8)
 (65) 公開番号 特開2003-264703 (P2003-264703A)
 (43) 公開日 平成15年9月19日 (2003.9.19)
 審査請求日 平成17年3月7日 (2005.3.7)

(73) 特許権者 000006747
 株式会社リコー
 東京都大田区中馬込1丁目3番6号
 (74) 代理人 100084250
 弁理士 丸山 隆夫
 (72) 発明者 山崎 由希子
 東京都大田区中馬込1丁目3番6号
 株式会社リコー内
 (72) 発明者 柳下 高弘
 東京都大田区中馬込1丁目3番6号
 株式会社リコー内
 審査官 堀井 啓明

最終頁に続く

(54) 【発明の名称】データ符号化装置、データ符号化方法、及びそのプログラム

(57) 【特許請求の範囲】

【請求項1】

シンボル列中、現符号化時点に隣接する、符号化済みのM個のシンボル列と、符号化前のM個のシンボル列と、が一致するか否かを判断し、

一致すると判断した場合に、前記符号化前のM個のシンボル列に、さらに、連続するM個のシンボル列が前記符号化済みのシンボル列と一致するか否かの判断を繰り返す比較手段と、前記比較手段により一致すると判断された一致回数をカウントするラン判定手段と、前記比較手段により一致すると判断された複数のシンボル列に代えて、前記Mの値と前記一致回数を符号化手段と、を有したデータ符号化装置であって、

さらに、所定の処理範囲の圧縮率を算出する圧縮率算出手段と、

10

前記圧縮率が目標値を達成しているか否かを判断する判断手段と、を有し、

前記判断手段により、圧縮率が目標値を達成していると判断されたか否かに応じて、前記比較手段により比較するシンボル数の上限値を変える、ことを特徴とするデータ符号化装置。

【請求項2】

前記判断手段によって、

圧縮率が目標値を達成していないと判断された場合には、前記比較手段が比較するシンボル数の上限値を上げ、

圧縮率が目標値を達成していると判断された場合には、前記比較手段が比較するシンボル数の上限値を下げる特徴とする請求項1記載のデータ符号化装置。

20

【請求項 3】

前記シンボル列を符号化対象データから作成するシンボル作成手段を有し、
前記符号化対象データの持つ周期に基づいて、1シンボルあたりのビット数を決定することを特徴とする請求項1又は2記載のデータ符号化装置。

【請求項 4】

前記シンボル作成手段は、
前記符号化対象データの持つ周期に基づいて、符号化対象データから複数ビットのデータを切り出す切り出し手段と、
切り出し手段により切り出されたデータをMoveToFront法により変換して前記シンボルを得る変換手段と、からなることを特徴とする請求項3記載のデータ符号化装置。 10

【請求項 5】

前記シンボル作成手段は、
前記符号化データから周期を検出する周期検出手段をさらに有し、
前記切り出し手段は、当該検知された周期に基づいて、切り出しビット数をえることを特徴とする請求項4記載のデータ符号化装置。

【請求項 6】

前記符号化手段は、
前記ラン判定手段によりカウントされた前記一致回数が所定の回数以上である場合には
、前記Mの値と前記一致回数とを符号化し、
前記ラン判定手段によりカウントされた前記一致回数が所定の回数未満である場合には
、前記符号化前のシンボル列を符号化することを特徴とする請求項1から5のいずれか1項記載のデータ符号化装置。 20

【請求項 7】

前記符号化手段は、
符号化しようとしている前記Mの値と前記一致回数と前記一致しないと判断された最初のシンボルと、直前に符号化した前記Mの値と前記一致回数と前記一致しないと判断された最初のシンボルと、において前記一致回数が等しい場合には、前記一致回数を省略した符号化を行うことを特徴とする請求項1から6のいずれか1項記載のデータ符号化装置。

【請求項 8】

シンボル列中、現符号化時点に隣接する、符号化済みのM個のシンボル列と、符号化前のM個のシンボル列と、が一致するか否かを判断し、
一致すると判断した場合に、前記符号化前のM個のシンボル列にさらに、連続するM個のシンボル列が前記符号化済みのM個のシンボル列と一致するか否かの判断を繰り返す比較ステップと、前記比較手段により一致すると判断された一致回数をカウントするラン判定ステップと、前記比較手段により一致すると判断された複数のシンボル列に代えて、前記Mの値と前記一致回数とを符号化する符号化ステップと、を有したデータ符号化方法であって、

さらに、所定の処理範囲の圧縮率を算出する圧縮率算出ステップと、
前記圧縮率が目標値を達成しているか否かを判断する判断ステップと、
前記判断ステップにより、圧縮率が目標値を達成していると判断されたか否かに応じて
、前記比較ステップにより比較するシンボル数の上限値をえるステップと、を有することを特徴とするデータ符号化方法。 40

【請求項 9】

シンボル列中、現符号化時点に隣接する、符号化済みのM個のシンボル列と、符号化前のM個のシンボル列と、が一致するか否かを判断し、
一致すると判断した場合に、前記符号化前のM個のシンボル列に、さらに、連続するM個のシンボル列が前記符号化済みのシンボル列と一致するか否かの判断を繰り返す比較処理と、前記比較手段により一致すると判断された一致回数をカウントするラン判定処理と、前記比較手段により一致すると判断された複数のシンボル列に代えて、前記Mの値と前記一致回数を符号化する符号化処理と、をコンピュータに実行させるデータ符号化プログ 50

ラムであって、

さらに、所定の処理範囲の圧縮率を算出する圧縮率算出処理と、
前記圧縮率が目標値を達成しているか否かを判断する判断処理と、
前記判断処理により、圧縮率が目標値を達成していると判断されたか否かに応じて、前記比較処理により比較するシンボル数の上限値を変える処理と、をコンピュータに実行させることを特徴とするデータ符号化プログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、データ符号化装置、データ符号化方法、及びそのプログラムに関し、具体的にはLZ77、およびLZ88のような辞書ベース方式を元にした圧縮技術を基に、特に画像データを効率良く圧縮するデータ符号化装置、データ符号化方法、及びそのプログラムに関する。 10

【0002】

【従来の技術】

近年、スキャナ、プリンタ、デジタルカメラ等の画像を扱う分野では、高解像度化が進み画像1枚あたりにおけるデータ量、いわゆる画素数は膨大なサイズになってきている。その膨大なデータを効率よくネットワーク上で転送したり、ハードディスク、メモリ等の記憶媒体に保存するために、処理速度、圧縮率において効率のよい圧縮方法が求められている。 20

【0003】

データを効率よく圧縮する方法として、ユニバーサル符号により圧縮する方法が実用化されている。このユニバーサル符号は、情報保存型のデータ圧縮方法であり、データ圧縮時に情報源の統計的な性質を予め仮定しないため、種々のタイプ(文字コード、オブジェクトコードなど)のデータに適用することが出来る。文書画像では、文字の輪郭等や文字間隔に類似性があり、又、網点画像は網点周期性、網点形状の同一性等が類似している。この類似性の持つ冗長性をユニバーサル符号により削減し、有効な圧縮を行うことが出来る。

【0004】

ユニバーサル符号の代表的な方法として、ジブ-レンペル(Ziv-Lempel)符号がある。Ziv-Lempel符号では、ユニバーサル型と、増分分解型(Incremental parsing)の2つのアルゴリズムが提案されており、ユニバーサル型アルゴリズムを用いた実用的な方法として、LZSS符号があり、又、増分分解型アルゴリズムを用いた実用的な方法として、LZW(Lempel-Ziv-Welch)符号がある。 30

【0005】

LZSSのベースとなるLZ77のアルゴリズムは、符号化データを過去のデータ系列の任意の位置から一致する最大長の系列に区切り、過去の系列の複製として符号化する方法である。具体的には、符号化済みの入力データを格納する移動窓と、これから符号化するデータを格納する先読みバッファとを備え、先読みバッファのデータ系列と移動窓のデータ系列のすべての部分系列とを照合して、移動窓中で一致する最大長の部分系列を求める。 40

【0006】

そして、移動窓中でこの最大長の部分系列を指定するために、「その最大長の部分系列の開始位置」と「一致する長さ」と「不一致をもたらした次のシンボル」との組を符号化する。次に、先読みバッファ内の符号化したデータ系列を移動窓に移して、先読みバッファ内に符号化したデータ系列分の新たなデータ系列を入力する。以下、同様の処理を繰り返していくことで、データを部分系列に分解して符号化を実行していくのである。

【0007】

一般的に、LZSS符号化は、符号化済の入力データを保存している移動窓数分、入力データ列との最長一致を検出する必要があるため、演算量が多いが高压縮率が得られるとい 50

われている。

【0008】

また、LZW符号化は、書き換え可能な辞書を設け、入力文字列を相異なる文字列に分け、この文字列を出現した順に番号を付けて辞書に登録すると共に、現在入力している文字列を辞書に登録してある最長一致文字列の辞書番号だけで表して符号化するというものである。このLZW符号化によれば、圧縮率はLZSS符号化より劣るが、シンプルで、計算も容易で、高速処理が出来ることから記憶装置のファイル圧縮、データ伝送などで使われるようになっている。

【0009】

また、特開平11-168389号公報「データ圧縮装置」によれば、移動窓数分の比較器を並列に持つのはハード負荷が大きいため、相関のありそうな、場所（真上、左など）に対する比較器のみに限定することで、処理速度、及びハード規模を考慮した方法が提案されている。

10

【0010】

また、特開平9-181610号公報「パターン圧縮方式及び装置」によれば、LZ法などで有効に圧縮できない長大な繰り返しパターンを少ない演算量で圧縮するというもので、LZ圧縮されたデータから低周波成分を抽出し、間引きし、自己相関を算出し、それに基づきパターンマッチングを行い、繰り返しを検出するという方法が提案されている。

【0011】

【発明が解決しようとする課題】

20

移動窓を使った方式であるLZSSのような過去の符号化済みのデータ中から一致するデータ列を検索する場合、窓サイズが大きいほど一致精度が上がるが、その分比較に要する処理が多くなり、処理速度、ハード規模に負担がかかってしまう。

【0012】

また、上記特開平11-168389号公報の方式では、比較処理は少なくて済むが、一致度の高い場所を保存しておくことにより結局、窓サイズとしてのバッファサイズが大きくなってしまう。また、上記特開平9-181610号公報の方式では、バッファを超えててしまうような長大な繰り返しパターンを発見するために、この方式は多数の工程を必要とするため、処理速度の点で問題があるといえる。

30

【0013】

本発明の方式では、窓サイズを可能な限り少なく設定することにより、比較処理も少なくて済み、処理速度の向上、及びハード規模を抑えるという利点がある。また、比較対象単位（符号化単位）のサイズを大きくすることで、長大なパターンを短いパターンに変換することが可能となり、繰り返しパターンを短い距離で発見することが可能となる。

【0014】

本発明は、符号化対象データの持つ周期がシンボル列の繰り返しとして発生することを利用し、それをランレンジス符号化することで効率よく圧縮する。また、これから符号化するシンボル列の直前の符号化済データを比較対象とすることで、比較範囲を小さくすることが可能となり、処理速度の向上、及びハード規模を小さくすることが可能となるデータ符号化装置、データ符号化方法、及びそのプログラムを提供することを第1の目的とする。

40

【0015】

また、本発明は、例えばディザ等の周期を発生させる処理が施されたデータが符号化対象データの場合、この周期に基づいた符号化単位にしておくことにより、元画像の周期を得るに当たり効率良く周期を得ることが出来、すなわち比較範囲をより小さくすることが可能となり、処理速度の向上、及びハード規模を小さくことが可能となるデータ符号化装置、データ符号化方法、及びそのプログラムを提供することを第2の目的とする。

【0016】

また、本発明は、符号化対象データの途中で周期が変わらるような場合（例えば画像データの場合、ディザの混在など）にも対応することが可能となるデータ符号化装置、データ符

50

号化方法、及びそのプログラムを提供することを第3の目的とする。

【0017】

また、本発明は、連続数に下限のしきい値を設けることで、連続数を符号化する際、圧縮率の低下を防ぎ符号化効率を上げることが可能となるデータ符号化装置、データ符号化方法、及びそのプログラムを提供することを第4の目的とする。

【0018】

また、本発明は、連続数に上限のしきい値を設けることで、連続数を符号化する際、符号のビット長を制限することが出来、符号化効率を上げることが可能となるデータ符号化装置、データ符号化方法、及びそのプログラムを提供することを第5の目的とする。

【0019】

また、本発明は、連続する事象が再帰的に発生した場合、これについて異なる符号化を行うことで圧縮効率を上げることが可能となるデータ符号化装置、データ符号化方法、及びそのプログラムを提供することを第6の目的とする。

【0020】

また、本発明は、符号化を進めた結果、目標とする所定の圧縮率に対して差が大きい場合、最適な処理がなされているとはいえない。このような不具合の対策として比較範囲を適応的に変化させることで圧縮率、処理速度の面で最適な圧縮効果を上げることが可能となるデータ符号化装置、データ符号化方法、及びそのプログラムを提供することを第7の目的とする。

【0021】

また、本発明は、符号化を進めた結果、ある所定の範囲における圧縮率が目標とする所定の圧縮率に対して所定値を満たしていない場合、比較範囲を広げることにより、連続するパターンの発生確立を上げ、圧縮効率を上げることが可能となるデータ符号化装置、データ符号化方法、及びそのプログラムを提供することを第8の目的とする。

【0022】

また、本発明は、符号化を進めた結果、ある所定の範囲における圧縮率が目標とする所定の圧縮率に対して所定値を満たしていた場合、その比較範囲は十分であるといえる。このとき更に比較範囲を狭くすることにより、比較処理を高速化することが可能となるデータ符号化装置、データ符号化方法、及びそのプログラムを提供することを第9の目的とする。

。

【0023】

【課題を解決するための手段】

上記目的を達成するために、請求項1記載の発明は、シンボル列中、現符号化時点に隣接する、符号化済みのM個のシンボル列と、符号化前のM個のシンボル列と、が一致するか否かを判断し、一致すると判断した場合に、前記符号化前のM個のシンボル列に、さらに、連続するM個のシンボル列が前記符号化済みのシンボル列と一致するか否かの判断を繰り返す比較手段と、前記比較手段により一致すると判断された一致回数をカウントするラン判定手段と、前記比較手段により一致すると判断された複数のシンボル列に代えて、前記Mの値と前記一致回数を符号化する符号化手段と、を有したデータ符号化装置であつて、さらに、所定の処理範囲の圧縮率を算出する圧縮率算出手段と、前記圧縮率が目標値を達成しているか否かを判断する判断手段と、を有し、前記判断手段により、圧縮率が目標値を達成していると判断されたか否かに応じて、前記比較手段により比較するシンボル数の上限値を変える、ことを特徴とするデータ符号化装置である。

また、請求項2記載の発明は、請求項1記載のデータ符号化装置において、前記判断手段によって、圧縮率が目標値を達成していないと判断された場合には、前記比較手段が比較するシンボル数の上限値を上げ、圧縮率が目標値を達成していると判断された場合には、前記比較手段が比較するシンボル数の上限値を下げる特徴とする。

【0024】

請求項3記載の発明は、請求項1又は2記載のデータ符号化装置において、前記シンボル列を符号化対象データから作成するシンボル作成手段を有し、前記符号化対象データの

10

20

30

40

50

持つ周期に基づいて、1シンボルあたりのビット数を決定することを特徴とする。

【0025】

請求項4記載の発明は、請求項3記載のデータ符号化装置において、前記シンボル作成手段は、前記符号化対象データの持つ周期に基づいて、符号化対象データから複数ビットのデータを切り出す切り出し手段と、切り出し手段により切り出されたデータをMoveToFront法により変換して前記シンボルを得る変換手段と、からなることを特徴とする。

【0026】

請求項5記載の発明は、請求項4記載のデータ符号化装置において、前記シンボル作成手段は、前記符号化データから周期を検出する周期検出手段をさらに有し、前記切り出し手段は、当該検知された周期に基づいて、切り出しビット数を変えることを特徴とする。 10

【0027】

請求項6記載の発明は、請求項1から5のいずれか1項記載のデータ符号化装置において、前記符号化手段は、前記ラン判定手段によりカウントされた前記一致回数が所定の回数以上である場合には、前記Mの値と前記一致回数とを符号化し、前記ラン判定手段によりカウントされた前記一致回数が所定の回数未満である場合には、前記符号化前のシンボル列を符号化することを特徴とする。

【0028】

請求項7記載の発明は、請求項1から6のいずれか1項記載のデータ符号化装置において、前記符号化手段は、符号化しようとしている前記Mの値と前記一致回数と前記一致しないと判断された最初のシンボルと、直前に符号化した前記Mの値と前記一致回数と前記一致しないと判断された最初のシンボルと、において前記一致回数が等しい場合には、前記一致回数を省略した符号化を行うことを特徴とする。 20

【0031】

請求項8記載の発明は、シンボル列中、現符号化時点に隣接する、符号化済みのM個のシンボル列と、符号化前のM個のシンボル列と、が一致するか否かを判断し、一致すると判断した場合に、前記符号化前のM個のシンボル列にさらに、連続するM個のシンボル列が前記符号化済みのM個のシンボル列と一致するか否かの判断を繰り返す比較ステップと、前記比較手段により一致すると判断された一致回数をカウントするラン判定ステップと、前記比較手段により一致すると判断された複数のシンボル列に代えて、前記Mの値と前記一致回数とを符号化する符号化ステップと、を有したデータ符号化方法であって、さらに、所定の処理範囲の圧縮率を算出する圧縮率算出ステップと、前記圧縮率が目標値を達成しているか否かを判断する判断ステップと、前記判断ステップにより、圧縮率が目標値を達成していると判断されたか否かに応じて、前記比較ステップにより比較するシンボル数の上限値を変えるステップと、を有することを特徴とするデータ符号化方法である。 30

【0032】

請求項9記載の発明は、シンボル列中、現符号化時点に隣接する、符号化済みのM個のシンボル列と、符号化前のM個のシンボル列と、が一致するか否かを判断し、一致すると判断した場合に、前記符号化前のM個のシンボル列に、さらに、連続するM個のシンボル列が前記符号化済みのシンボル列と一致するか否かの判断を繰り返す比較処理と、前記比較手段により一致すると判断された一致回数をカウントするラン判定処理と、前記比較手段により一致すると判断された複数のシンボル列に代えて、前記Mの値と前記一致回数を符号化する符号化処理と、をコンピュータに実行させるデータ符号化プログラムであって、さらに、所定の処理範囲の圧縮率を算出する圧縮率算出処理と、前記圧縮率が目標値を達成しているか否かを判断する判断処理と、前記判断処理により、圧縮率が目標値を達成していると判断されたか否かに応じて、前記比較処理により比較するシンボル数の上限値を変える処理と、をコンピュータに実行させることを特徴とするデータ符号化プログラムである。 40

【0034】

【発明の実施の形態】

以下、本発明の実施の形態を添付図面を参照しながら詳細に説明する。

【0035】

本発明の第1の実施の形態について説明する。まず、基本構成を図1のブロック図に示す。図1の入力データはテキストデータでもバイナリデータでもよいが、ここでは1bit/画素のディザ処理等の中間調処理された画像データを入力データとした例を示す。まず、入力データを複数ビット切り出し部101において、数ビットをかためて1データとして切り出す。次に中間データ作成部102においてMTF(MoveToFront)等の手法により各データをNビットのデータに変換して中間データとして出力する。

【0036】

ここで、中間データに変換する意味は入力データの冗長性を取り除くという理由である。例えば、複数ビット切り出し部101において切り出すかたまりが比較的小さい場合、例えれば4ビット程度だったとき、切り出されたデータの発生パターンは、たかだか16通りである。このような場合はよいが、例えば切り出すビット数が20ビットだった場合は2の20乗もの種類になってしまう。

10

【0037】

こうなると発生パターンが膨大になり、ソフト上の処理速度的にもハード規模においても支障が生じる。しかしながら、ディザ処理が施された画像データにはもともと周期性がある。ディザのもつ周期が20ビットだったとして、それに基づいて20ビットを1データ化すると、その分布はかなり偏りのあるものとなるはずである。それを利用して、MTF(MoveToFront)で20ビットのデータを比較的小さいビット数のNビットに変換して中間データとして出力する。

20

【0038】

MTFについて、図2を参照して説明する。ここではMTFに入力されるデータは8ビットの文字データであるとする。まず、図2の1のように入力シンボル”A”があったとき、”A”が格納されている辞書のインデックス番号”2”を出力値とする変換を行う。次に、2のように入力シンボルの”A”を辞書の先頭に移動して、辞書を更新する。

【0039】

これによって処理が進むにつれ、よく出現するシンボルが小さいインデックス番号の位置に集まるので、任意の分布形状を持った入力を、0をピークとする分布に変更することが出来る。この辞書の登録数は、任意の数に制限出来る。登録されている以外のシンボルが入力された場合、”ESC符号+生データ”として出力し、そのシンボルを新規に辞書の先頭に登録することで対応出来る。

30

【0040】

図1に戻り、MTFなどの処理が施された中間データは比較器103に入力される(以下、この中間データを文字と呼ぶ)。比較器103について、図3、図4を参照して説明する。バッファのサイズを16文字分($n = 16$)とし、すでに8文字分が符号化されているものとする(図3a)。バッファ中の符号化開始位置(図の点線の位置)から始まる文字(点線の右側)と、バッファ中の符号化済みの文字列の符号化開始位置の直前点線の左側)の8種類の長さの文字列とを比較して最長に一致するものを検索する(ステップS1)。

40

【0041】

この場合”abc”が最長一致と判定されるが(ステップS1/YES)、ここでは符号を出力せずに一時、この先頭アドレス、すなわち文字列長を表す”3”を一時記憶しておく。次に、この一致した文字列3文字分が符号化済みのバッファ部分の最後尾にくるように、バッファ中の文字を左にシフトし、符号化開始位置をずらす(図3b)。次の符号化開始位置をみると先ほどと同様に”abc”的文字列が最長一致し、再度アドレス”3”が判定される。

【0042】

ここで、図1に示すラン判定部104において、先ほど一時保存されているアドレスと比較し、同値であることが分かる(ステップS2/YES)。ここでランを+1する(ステ

50

ップS3)。ここでもまだ符号を出力しない。次に先の説明と同様に3文字分左にシフトする(ステップS5)。今度の符号化開始文字列と符号化済み文字列とは一致する文字列がない(ステップS1/N0)。このように、一致しなかったときにはじめて今までのランと現在の一一致しなかった文字を符号として出力する(ステップS8)。この時の符号は”abc”的開始位置、すなわち文字の長さである”3”と、ランの”2”、そして”d”的符号が出力される(図3c)。以上の比較器103による処理の詳細は、図4のフローチャートに示した。

【0043】

このように圧縮効率を上げるためにランを発生させたい。すなわち文字列が繰り返されるように、元画像の持つ周期を利用して、それに基づいて中間データを作成する。例えば符号化対象データが8bit/画素のデータで、これに 20×20 のディザマトリクスサイズのディザが施されて1bit/画素に中間調処理されていたとする。このとき中間調処理後のデータはX方向に20の周期で類似したデータが繰り返されることが予想出来る。この”20”という周期が既知である場合、図1における複数ビット切り出し部101は、入力データを20ビット単位に切り出して中間データ作成部102に渡す。これにより、切り出されたデータに類似性が増すので、比較器103において一致精度を向上させることが出来る。

【0044】

また、上記符号化対象データが持つ周期を自動的に検出する。図5に処理ブロック図を示す。図1に示した処理の先頭に周期検出部501が追加される。図6に周期の検出方法の一例を示す。このような自動周期検出部501を持ってば、その判定範囲によっては1ライン中でも異なる周期を発見することが出来る。例えば、画像処理によってはイメージ部分とグラフィック部分とが異なるディザを施される場合がある。すなわち異なる周期が1ライン中に発生しても対応することが可能となる。

【0045】

以上のような符号化対象データの周期を利用した切りだしは圧縮効果を上げるものであるが、もちろんこのような処理をしなくても所定の効果は得られるものである。

【0046】

次に第2の実施の形態について説明する。第1の実施の形態に示した例において、図1のラン判定部104で、ランについてのしきい値を設けるというものである。図6に示したように、ランを作成しない場合、“ab”という文字列があったとして、“a”が3ビット、“b”が4ビットの符号が割り当てられていたとすると、“ab”で7ビットとなる。

【0047】

ランの符号が例えば、ESC2符号=5ビット、アドレス=5ビット、ラン長=8ビットだとすると18ビット使用することになる。ランを作成する場合、通常の1文字単位の符号化よりも圧縮効率をあげることが前提となるため、この例では”ab”が2回で14ビット、3回で21ビットとなるため、3回以上の場合でないとランを作成する効果が得られないことになる。よってランの最低値となるしきい値(A)が必要となる。

【0048】

次、ラン長の上限のしきい値(B)を設ける。すなわち、ラン長を表すビット数を制限するというものである。一般にバイナリデータが圧縮対象のランレンジス符号化は、ラン長が長ければ長いほど圧縮効率が上がるが、今回のようなランの符号としてのビット長が固定である場合、ラン長を表す符号のビット長はある程度制限する必要がある。なぜなら実際に短いラン長しか発生しなかった場合に無駄なビットを出力することになるからである。

【0049】

以上のようなラン長の下限のしきい値は、上記の例ではランの値のみから判断したが、実際発生するビット数は、文字列を構成する各文字のビット長とランの個数によって決定するのでそれらをすべて考慮して圧縮効率が低下しないしきい値を設定することも出来る。

10

20

30

40

50

【0050】

次に、第3の実施の形態について図7を参照して説明する。図7のような符号化対象データがバッファ中にあるとする。このデータを第1の実施の形態に示した処理で符号化すると、"Aの符号 + ESC2符号 + アドレス(1) + ラン(3)"に続いて、"Bの符号 + ESC2符号 + アドレス(1) + ラン(3)"、"Cの符号 + ESC2符号 + アドレス(1) + ラン(3)"と出力されることになる。

【0051】

しかしながら、この3種類の符号は見て分かるように構成は全く同じであって、文字の種類だけが異なるものである。これを効率良く符号化するために、本実施の形態では"BBB"の出力は"Bの符号 + ESC3符号"に置き換える。同様に次の"CCC"は"Cの符号 + ESC3符号"とする。このようにランを2段階作成することで、更に圧縮効率を向上させることが出来る。ここで例は1文字の連続をあげたが、"ABCABCABCDEFDEFDEF"のような3文字がN回繰り返される様な場合にも適応できる。10

【0052】

次に、第4の実施の形態について説明する。本実施の形態は、図1における比較器103において、比較処理におけるバッファサイズがある基準を元に適応的に変化させるというものである。図8にフローチャートを示す。ある所定の処理範囲を設定し、その処理範囲が終了したら(ステップS23/YE S)そこまでの符号化効率として圧縮率(M)を算出し(ステップS24)、ある所定の目標圧縮率を達成しているか否かを判定し(ステップS25)、達成していない場合に(ステップS25/NO)、比較範囲(=窓サイズ)を広げる(ステップS26)。20

【0053】

通常のLZ方式等で持つ窓サイズ、特に符号化済みのデータを格納するバッファはかなり大きなサイズを必要とする。これは圧縮効率を上げるためにあるが、このサイズによっては処理速度や回路規模に多大な影響を与える。

【0054】

本発明において第1の実施の形態から説明に使用したバッファサイズはもともと処理速度、及び回路規模を満たすのに十分な程度の極少ないサイズを想定し、局所的な繰り返しを抽出することを目的としている。したがって、実際に窓サイズよりも長い周期の繰り返しが存在していた場合、これを検出できない場合もある。これを回避するために、窓サイズを広げて処理を進める。広げるサイズにも上限を設定することで処理速度の低下を抑えることが出来る。30

【0055】

また、上記目的とは逆に、目標圧縮率が達成されている場合、窓サイズを小さくする。小さくすることで、比較器103の処理速度を向上させる効果がある。こちらも同様に処理速度とのバランスで必要以上に小さくする必要はなく、圧縮効率を低下させない程度の制限を設定しておくことも出来る。

【0056】

なお、上述した実施の形態は、本発明の好適な実施の形態の一例を示すものであり、本発明はそれに限定されることなく、その要旨を逸脱しない範囲内において、種々変形実施が可能である。40

【0057】

なお、本発明はコンピュータにプログラムを実行させることにより実現可能である。当該プログラムは、光記録媒体、磁気記録媒体、光磁気記録媒体もしくは半導体IC記録媒体に記録されて提供されるか、またはプログラムサーバからFTP、HTTP等のプロトコルによりネットワークを介してダウンロードされて提供される。

【0058】**【発明の効果】**

以上の説明から明らかなように、本発明によれば次のような効果が得られる。符号化対象データの持つ周期がデータ列の繰り返しとして発生することを利用し、それをランレンジ50

ス符号化することで効率よく圧縮することが出来る。また、これから符号化するシンボル列の直前の符号化済データを比較対象として、比較範囲を小さくすることにより、処理速度の向上、及びハード規模を小さくすることが出来る。

【0059】

また、本発明によれば、例えば、ディザ等の周期を発生させる処理が施されたデータが符号化対象データの場合、この周期に基づいた符号化単位にしておくことにより、元画像の周期を得るに当たり効率良く周期を得ることが出来、すなわち比較範囲をより小さくすることが可能となり、処理速度の向上、及びハード規模を小さくすることが出来る。

【0060】

また、本発明によれば、符号化対象データの途中で周期が変わらるような場合（たとえば画像データの場合、ディザの混在など）にも対応することが出来る。

10

【0061】

また、本発明によれば、連続数に下限のしきい値を設けることで、連続数を符号化する際、圧縮率の低下を防ぎ符号化効率を上げることが出来る。

【0062】

また、本発明によれば、連続数に上限のしきい値を設けることで、連続数を符号化する際、符号のbit長を制限することが出来、符号化効率を上げることが出来る。

【0063】

また、本発明によれば、連続する事象が再帰的に発生した場合、これについて、異なる符号化を行うことで圧縮効率を上げることが出来る。

20

【0064】

また、本発明によれば、符号化を進めた結果、目標とする所定の圧縮率に対して差が大きい場合、最適な処理がなされているとはいえない。このような不具合の対策として比較範囲を適応的に変えることで、圧縮率、処理速度の面で最適な条件を得ることが出来る。

【0065】

また、本発明によれば、符号化を進めた結果、ある所定の範囲における圧縮率が目標とする所定の圧縮率に対して所定値を満たしていない場合、比較範囲を広げることにより、連続するパターンの発生確立を上げることが出来、圧縮効率を上げることが出来る。

【0066】

また、本発明によれば、符号化を進めた結果、ある所定の範囲における圧縮率が目標とする所定の圧縮率に対して所定値を満たしていた場合、その比較範囲は十分であるといえる。このとき更に比較範囲を狭くすることにより比較処理を削減出来るので、比較処理を高速化することが出来る。

30

【図面の簡単な説明】

【図1】本発明の基本構成を示すブロック図である。

【図2】MTFを説明するための図である。

【図3】比較器103を説明するための図である。

【図4】比較器103の処理を説明するためのフローチャートである。

【図5】本発明のバリエーションの構成を示すブロック図である。

【図6】周期を自動検出するための一構成例を示す図である。

40

【図7】第3の実施の形態を説明するための図である。

【図8】第4の実施の形態を説明するためのフローチャートである。

【符号の説明】

101 複数ビット切り出し部

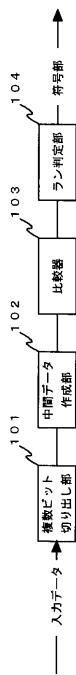
102 中間データ作成部

103 比較器

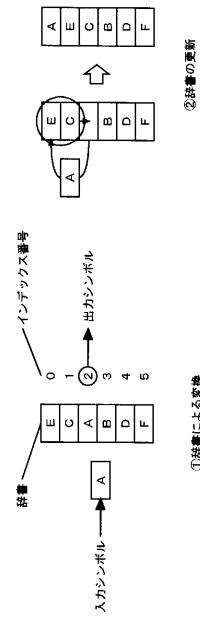
104 ラン判定部

501 周期検出部

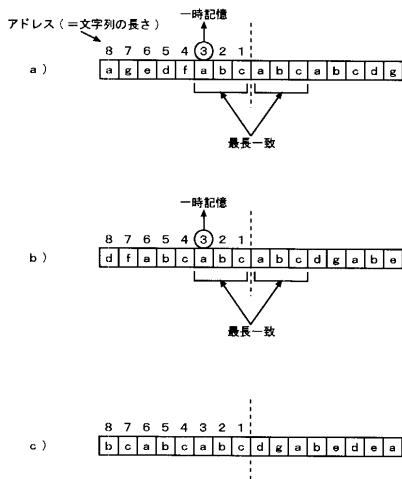
【図1】



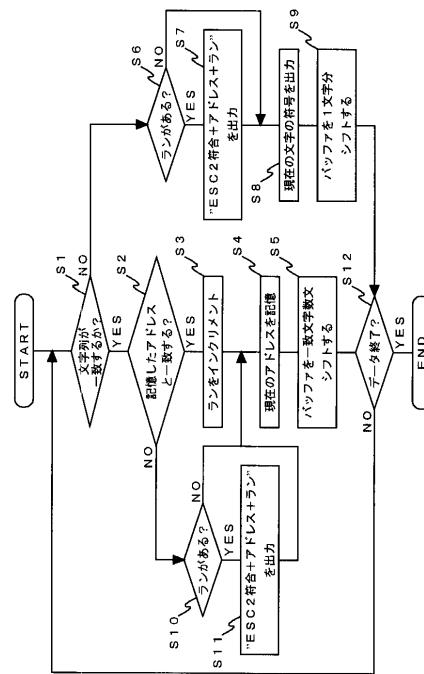
【図2】



【図3】



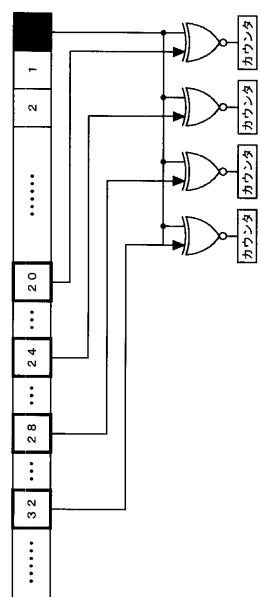
【図4】



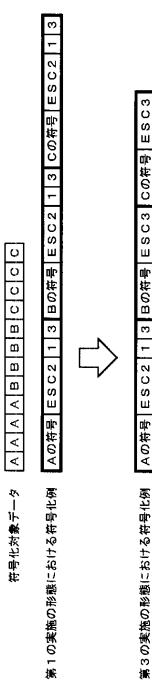
【図5】



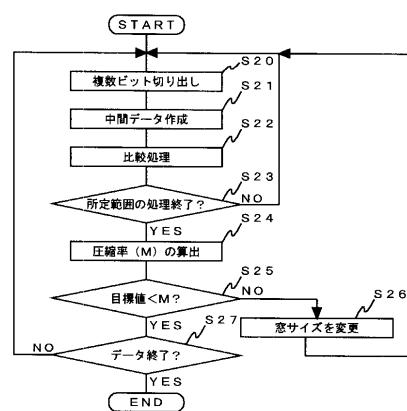
【図6】



【図7】



【図8】



フロントページの続き

(56)参考文献 特開2003-174564(JP,A)

特開平11-168389(JP,A)

特開平09-214353(JP,A)

特開平3-239069(JP,A)

(58)調査した分野(Int.Cl., DB名)

H04N1/41-1/419

H03M3/00-11/00