(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平8－204577

(43)公開日　平成８年(1996)８月９日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｈ０３Ｍ　7/36		9382－5Ｋ		
Ａ６３Ｆ　9/22		Ｈ		
Ｇ０６Ｆ　5/00		Ｈ		
Ｇ１０Ｌ　9/18		Ｂ		
		Ｈ		

審査請求　未請求　請求項の数13　ＯＬ　（全 20 頁）

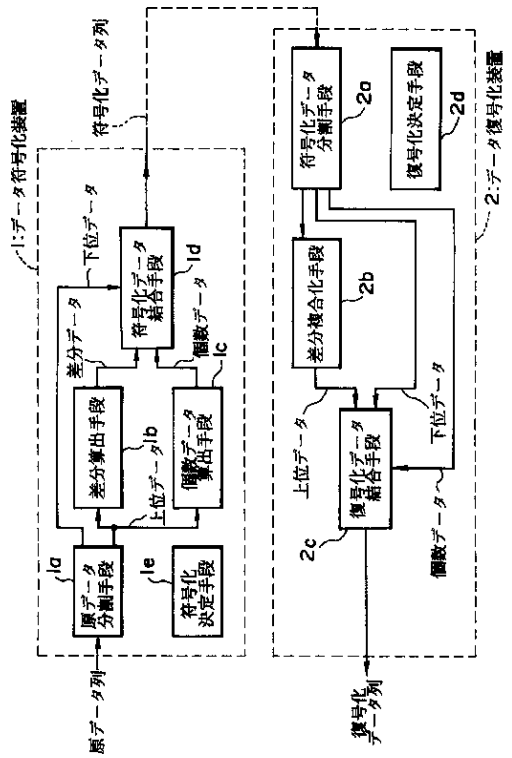
(21)出願番号　特願平7－10310

(22)出願日　平成７年(1995)１月26日

(71)出願人　000132471
株式会社セガ・エンタープライゼス
東京都大田区羽田１丁目２番12号
(72)発明者　河内　哲也
東京都大田区羽田１丁目２番12号　株式会社セガ・エンタープライゼス内
(74)代理人　弁理士　稲葉　良幸　（外２名）

(54)【発明の名称】　データ符号化装置、データ復号化装置、データ符号化方法、および、データ復号化方法

(57)【要約】
【目的】　符号化誤差を最小限に抑えながら、簡易なアルゴリズムで原データの符号化・復号化を行う。
【構成】　音声データ等を表す原データを、上位データ、下位データに分割する原データ分割手段１aを設ける。また、上位データ間の差分データを算出する差分算出手段１ｂ、原データの個数を表す個数データを求める個数データ算出手段１ｃを設ける。符号化決定手段１ｄにより、下位データおよび差分データを結合する。このように、原データのうち、比較的に変化の少ない上位データの差分データを算出することにより、符号化誤差を最小限に抑えながら、簡易なアルゴリズムで原データの符号化・復号化を行うことが可能となる。



1

【特許請求の範囲】

【請求項１】　複数ビットからなる原データが複数配列された原データ列を入力する入力手段と、
最小ビットから最大ビットにかけて桁上がりがなされる原データを、最大ビットを含む上位データ、および、最小ビットを含む下位データに分割する原データ分割手段と、
隣接し合う原データにおける、それぞれの上位データ間の差分データを算出する差分算出手段と、
上記下位データおよび上記差分データが配列された符号化データ列を出力する出力手段とを備えたデータ符号化装置。

【請求項２】　複数ビットからなる原データが複数配列された原データ列を入力する入力手段と、
最小ビットから最大ビットにかけて桁上がりがなされる原データを、最大ビットを含む上位データ、および、最小ビットを含む下位データに分割する原データ分割手段と、
隣接し合う原データにおける、それぞれの上位データ間の差分データを算出する差分算出手段と、
差分データが連続して所定範囲内に収まる原データの個数を算出し、個数データを出力するデータ数算出手段と、
上記個数データ、上記差分データ、および、上記下位データが配列された符号化データ列を出力する出力手段を備えたデータ符号化装置。

【請求項３】　上記下位データのうち、上記差分データのビット数に相当する下位ビットを切り捨てるとともに、この下位データに上記差分データを結合させる符号化データ結合手段を備えた請求項２記載のデータ符号化装置。

【請求項４】　上記所定範囲は＂０＂であり、かつ、差分データのビット数は＂０＂である請求項２記載のデータ符号化装置。

【請求項５】　符号化データ列を構成する符号化データの個数が、原データ列を構成する原データの個数よりも少なくなった場合に、上記出力手段から符号化データ列を出力する符号化決定手段を備えた請求項２記載のデータ符号化装置。

【請求項６】　請求項１記載の符号化データ列を入力する入力手段と、
符号化データ列を分割することにより、差分データおよび下位データを抽出する符号化データ分割手段と、
差分データに基づき上位データを生成する差分復号化手段と、
上位データおよび下位データを結合することにより復号化データを生成する復号化データ結合手段と、
複数の復号化データが配列された復号化データ列を出力する出力手段とを備えたデータ復号化装置。

【請求項７】　請求項２記載の符号化データ列を入力する入力手段と、
符号化データ列を分割することにより、差分データ、下位データ、個数データを抽出する符号化データ分割手段と、
差分データに基づき上位データを生成する差分復号化手段と、
個数データにより表された個数分、上位データおよび下位データを結合することにより、復号化データを生成する復号化データ結合手段と、
複数の復号化データが配列された復号化データ列を出力する出力手段とを備えたデータ復号化装置。

【請求項８】　原データ列を構成する原データを上位データおよび下位データに分割し、
隣接し合う原データにおける、それぞれの上位データ間の差分データを算出し、
差分データが連続して所定範囲内に収まる原データの個数を算出し、算出結果を個数データとして出力し、
上記個数データ、上記差分データ、および、上記下位データが配列された符号化データ列を生成するデータ符号化方法。

【請求項９】　上記下位データのうち、上記差分データのビット数に相当する下位ビットを切り捨てるとともに、この下位データに上記差分データを結合させる請求項８記載のデータ符号化方法。

【請求項１０】　複数ビットよりなる原データが複数配列された原データ列を以下の符号化モード(a)乃至(d)に従い符号化することにより、符号化データ列を生成するデータ符号化方法。

(a)　基準となる原データとの差が所定範囲に収まる他の原データが、所定個数以上連続して配列されている場合には、上記基準となる原データと、上記連続する他の原データの個数を表す個数データとが配列された符号化データ列を生成する。

(b)　符号化モード(a)の符号化処理を行わない場合であって、
隣接し合う原データ同士の差分データが所定範囲に収まり、かつ、このような原データが連続して所定個数以上配列されている場合には、
上記差分データと、上記原データの個数を表す個数データとが配列された符号化データ列を生成する。

(c)　符号化モード(a)および(b)の符号化処理を行わない場合であって、
隣接し合う原データの上位データ間の差を表す差分データが所定範囲内に収まり、かつ、このような原データが連続して所定個数以上配列されている場合には、
上記原データの個数を表す個数データと、上記原データの下位データと、上記差分データとが配列された符号化データ列を生成する。

(d)　符号化モード(a)、(b)、および、(c)の符号化処理を行わない場合には、原データが配列された符号化デー

タ列を生成する。
【請求項１１】　符号化処理の行われた符号化モード(a)乃至(d)のいずれかを示すデータを上記符号化データ列中に配列する請求項１０記載のデータ符号化方法。
【請求項１２】　請求項８または請求項９のいずれかに記載の符号化データ列を分割することにより、差分データおよび下位データを抽出し、
差分データに基づき上位データを生成し、
上位データおよび下位データを結合することにより復号化データを生成し、
生成された複数の復号化データが配列された復号化データ列を出力するデータ復号化方法。
【請求項１３】　請求項１１記載の符号化データ列から、符号化モード(a)乃至(d)のいずれかを示すデータを抽出し、
符号化モードに従い、以下の復号化処理を行うデータ復号化方法。
(1)　抽出されたデータが符号化モード(a)である場合には、
符号化データ列から、基準となる原データと、原データの個数を表す個数データとを抽出し、
個数データにより示された個数の基準データが配列された復号化データ列を出力する。
(2)　抽出されたデータが符号化モード(b)である場合には、
符号化データ列から、差分データと、原データの個数を表す個数データとを抽出し、
差分データに基づき復号化データを生成し、
個数データにより示された個数の復号化データが配列された復号化データ列を出力する。
(3)　抽出されたデータが符号化モード(c)である場合には、
符号化データ列から、差分データと、下位データと、原データの個数を表す個数データとを抽出し、
差分データに基づき上位データを生成し、
上位データと下位データとを結合することにより復号化データを生成し、
個数データにより示された個数の復号化データが配列された復号化データ列を出力する。
(4)　抽出されたデータが符号化モード(d)である場合には、
符号化データ列中から原データを抽出し、
抽出された原データが配列された復号化データ列を出力する。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】本発明は、データ符号化装置、データ復号化装置、データ符号化方法、および、データ復号化方法に関し、詳しくは音声データ、画像データ等の圧縮、伸張を行うデータ符号化装置、データ復号化装置、データ符号化方法、および、データ復号化方法に関する。
【０００２】
【従来の技術】ゲーム機器等のいわゆるマルチメディアの分野においては、音声データ、画像データ等をメモリカセット、ＣＤ－ＲＯＭ等に効率良く記録するために、音声データ等の圧縮を行うデータ符号化が用いられることが多い。このようなデータ符号化方法として、ΔＰＣＭ、ＤＰＣＭ、ＡＤＰＣＭ等の符号化方法が古くから知られている。これらの符号化方法は、基本的にはデータの差分を算出するものであり、比較的に簡易なアルゴリズムで圧縮伸張を行うことが可能なものである。
【０００３】また、近年になり、ＪＰＥＧ（Joint Photographic Coding Expert Group）、ＭＰＥＧ（Moving Picture Expert Group）等のデータ符号化方法が採用されるようになった。ＪＰＥＧ、ＭＰＥＧ等のデータ符号化方法は、直交関数変換であるＤＣＴ（離散コサイン変換）処理、量子化処理等よりなるものであり、比較的に効率のよい圧縮伸張を行うことが可能である。
【０００４】さらに、他のデータ符号化方法として、特開昭５８－１７１０９４号公報、特開昭５９－３５０４０号公報に開示されたものがある。特開昭５８－１７１０９４号公報に開示されたデータ符号化方法は、高次の展開式に展開した後、時間軸上における展開式の相関を考慮してデータ圧縮を行う方法である。また、特開昭５９－１７１０９４号公報に開示されたデータ符号化方法は、包絡線情報と波形ピッチとの２種類のデータを用いて波形データを表現することにより、データ圧縮を行うものである。
【０００５】
【発明が解決しようとする課題】しかしながら、上述した各種のデータ符号化方法等にあっては、以下の問題が生じていた。
【０００６】ΔＰＣＭ、ＤＰＣＭ、ＡＤＰＣＭのデータ符号化方法は、比較的に簡易なアルゴリズムにより原データの圧縮を行うことができるが、圧縮伸張に伴う音質劣化が大きいという問題を有している。例えば、原データの変化量が大きいような場合には、原データの差分が予め定められたデータ長を超えてしまう場合がある。この結果、復号化後のデータと原データとの間に誤差が生じ、この誤差は量子化ノイズとなって音質劣化、画質劣化の一因となる。
【０００７】また、ＪＰＥＧ、ＭＰＥＧ等のデータ符号化方法は比較的に高い圧縮率を実現できるという利点を有している。ところが、ＪＰＥＧ、ＭＰＥＧ等のデータ符号化方法は、ＤＣＴ等の複雑なアルゴリズムを必要とするため、処理が複雑となり、処理に長い時間を要する。
【０００８】さらに、特開昭５８－１７１０９４号公報に開示されたデータ符号化方法は、原データを高次の展

開式を算出しなければならず、符号化・復号化のアルゴリズムが複雑になってしまう。特開昭５９－３５０４０号公報に開示されたデータ符号化方法は、包絡線とピッチのみにより原データを表現するために、復号化後のデータと原データとの誤差が大きくなり易い。

【０００９】本発明は以上の課題に鑑みてなされたものであり、本発明の目的は、データ符号化装置、データ復号化装置、データ符号化方法、および、データ復号化方法において、簡易なアルゴリズムで符号化・復号化誤差の少ない圧縮伸張を行うことにある。

【００１０】

【課題を解決するための手段】請求項１記載の発明は、複数ビットからなる原データが複数配列された原データ列を入力する入力手段と、最小ビットから最大ビットにかけて桁上がりがなされる原データを、最大ビットを含む上位データ、および、最小ビットを含む下位データに分割する原データ分割手段と、隣接し合う原データにおける、それぞれの上位データ間の差分データを算出する差分算出手段と、上記下位データおよび上記差分データが配列された符号化データ列を出力する出力手段とを備えたデータ符号化装置である。

【００１１】請求項２記載の発明は、複数ビットからなる原データが複数配列された原データ列を入力する入力手段と、最小ビットから最大ビットにかけて桁上がりがなされる原データを、最大ビットを含む上位データ、および、最小ビットを含む下位データに分割する原データ分割手段と、隣接し合う原データにおける、それぞれの上位データ間の差分データを算出する差分算出手段と、差分データが連続して所定範囲内に収まる原データの個数を算出し、個数データを出力するデータ数算出手段と、上記個数データ、上記差分データ、および、上記下位データが配列された符号化データ列を出力する出力手段を備えたデータ符号化装置である。

【００１２】請求項３記載の発明は、上記下位データのうち、上記差分データのビット数に相当する下位ビットを切り捨てるとともに、この下位データに上記差分データを結合させる符号化データ結合手段を備えた請求項２記載のデータ符号化装置である。

【００１３】請求項４記載の発明は、上記所定範囲は"０"であり、かつ、差分データのビット数は"０"である請求項２記載のデータ符号化装置である。

【００１４】請求項５記載の発明は、符号化データ列を構成する符号化データの個数が、原データ列を構成する原データの個数よりも少なくなった場合に、上記出力手段から符号化データ列を出力する符号化決定手段を備えた請求項２記載のデータ符号化装置である。

【００１５】請求項６記載の発明は、請求項１記載の符号化データ列を入力する入力手段と、符号化データ列を分割することにより、差分データおよび下位データを抽出する符号化データ分割手段と、差分データに基づき上位データを生成する差分復号化手段と、上位データおよび下位データを結合することにより復号化データを生成する復号化データ結合手段と、複数の復号化データが配列された復号化データ列を出力する出力手段とを備えたデータ復号化装置である。

【００１６】請求項７記載の発明は、請求項２記載の符号化データ列を入力する入力手段と、符号化データ列を分割することにより、差分データ、下位データ、個数データを抽出する符号化データ分割手段と、差分データに基づき上位データを生成する差分復号化手段と、個数データにより表された個数分、上位データおよび下位データを結合することにより、復号化データを生成する復号化データ結合手段と、複数の復号化データが配列された復号化データ列を出力する出力手段とを備えたデータ復号化装置である。

【００１７】請求項８記載の発明は、原データ列を構成する原データを上位データおよび下位データに分割し、隣接し合う原データにおける、それぞれの上位データ間の差分データを算出し、差分データが連続して所定範囲内に収まる原データの個数を算出し、算出結果を個数データとして出力し、上記個数データ、上記差分データ、および、上記下位データが配列された符号化データ列を生成するデータ符号化方法である。

【００１８】請求項９記載の発明は、上記下位データのうち、上記差分データのビット数に相当する下位ビットを切り捨てるとともに、この下位データに上記差分データを結合させる請求項８記載のデータ符号化方法である。

【００１９】請求項１０記載の発明は、複数ビットよりなる原データが複数配列された原データ列を以下の符号化モード(a)乃至(d)に従い符号化することにより、符号化データ列を生成するデータ符号化方法である。

【００２０】(a)　基準となる原データとの差が所定範囲に収まる他の原データが、所定個数以上連続して配列されている場合には、上記基準となる原データと、上記連続する他の原データの個数を表す個数データとが配列された符号化データ列を生成する。

【００２１】(b)　符号化モード(a)の符号化処理を行わない場合であって、隣接し合う原データ同士の差分データが所定範囲に収まり、かつ、このような原データが連続して所定個数以上配列されている場合には、上記差分データと、上記原データの個数を表す個数データとが配列された符号化データ列を生成する。

【００２２】(c)　符号化モード(a)および(b)の符号化処理を行わない場合であって、隣接し合う原データの上位データ間の差を表す差分データが所定範囲内に収まり、かつ、このような原データが連続して所定個数以上配列されている場合には、上記原データの個数を表す個数データと、上記原データの下位データと、上記差分データとが配列された符号化データ列を生成する。

【００２３】(d)　符号化モード(a)、(b)、および、(c)の符号化処理を行わない場合には、原データが配列された符号化データ列を生成する。

【００２４】請求項１１記載の発明は、符号化処理の行われた符号化モード(a)乃至(d)のいずれかを示すデータを上記符号化データ列中に配列する請求項１０記載のデータ符号化方法である。

【００２５】請求項１２記載の発明は、請求項８または請求項９のいずれかに記載の符号化データ列を分割することにより、差分データおよび下位データを抽出し、差分データに基づき上位データを生成し、上位データおよび下位データを結合することにより復号化データを生成し、生成された複数の復号化データが配列された復号化データ列を出力するデータ復号化方法である。

【００２６】請求項１３記載の発明は、請求項１１記載の符号化データ列から、符号化モード(a)乃至(d)のいずれかを示すデータを抽出し、符号化モードに従い、以下の復号化処理を行うデータ復号化方法。

【００２７】(1)　抽出されたデータが符号化モード(a)である場合には、符号化データ列から、基準となる原データと、原データの個数を表す個数データとを抽出し、個数データにより示された個数の基準データが配列された復号化データ列を出力する。

【００２８】(2)　抽出されたデータが符号化モード(b)である場合には、符号化データ列から、差分データと、原データの個数を表す個数データとを抽出し、差分データに基づき復号化データを生成し、個数データにより示された個数の復号化データが配列された復号化データ列を出力する。

【００２９】(3)　抽出されたデータが符号化モード(c)である場合には、符号化データ列から、差分データと、下位データと、原データの個数を表す個数データとを抽出し、差分データに基づき上位データを生成し、上位データと下位データとを結合することにより復号化データを生成し、個数データにより示された個数の復号化データが配列された復号化データ列を出力する。

【００３０】(4)　抽出されたデータが符号化モード(d)である場合には、符号化データ列中から原データを抽出し、抽出された原データが配列された復号化データ列を出力する。

【００３１】

【作用】請求項１記載の発明において、入力手段は、原データが複数配列された原データ列を入力する。分割手段は、複数ビットからなる原データを上位データおよび下位データに分割し、差分算出手段は、隣接し合う原データにおける、それぞれの上位データ間の差分データを算出する。そして、出力手段は、下位データおよび差分データが配列された符号化データ列を出力する。

【００３２】本発明によれば、原データのうち、比較的に変化の少ない上位データのみの差分データを用いて符号化することにより、通常の差分ＰＣＭに比べて差分データのデータ量を低減することができる。すなわち、ＴＶゲーム機等のマルチメディアにおいて使用される音声データ、画像データは、一般に上位データの変化量が比較的に少ない。このため、上位データの差分データのデータ量を低減することができ、高圧縮の符号化・復号化を実現することができる。さらに、本発明にあっては、比較的に変化の少ない上位データの差分を算出するため、差分データが予め定められたデータ長を超えることにより生じる符号化・復号化誤差を低減することができる。

【００３３】また、上位データのみの差分を算出すればよいため、符号化・復号化アルゴリズムを簡略化することができる。このため、例えば、家庭用ＴＶゲーム機等においては、装置の製造コストを低減することが可能となる。さらに、簡易なアルゴリズムで符号化・復号化を実現できるため、高速の符号化・復号化を行うことができる。

【００３４】請求項２記載の発明において、入力手段は、原データが複数配列された原データ列を入力する。分割手段は、複数ビットからなる原データを上位データおよび下位データに分割し、差分算出手段は、隣接し合う原データにおける上位データ間の差分データを算出する。そして、データ数算出手段は、差分データが連続して所定範囲内に収まる原データの個数を算出し、出力手段は、個数データ、差分データ、および、下位データが配列された符号化データ列を出力する。

【００３５】本発明によれば、比較的に変化の少ない上位データのみの差分データを用いて符号化することにより、符号化誤差を低減しながら高圧縮の符号化を行うことができる。また、上位データのみの差分を算出すればよいため、符号化・復号化アルゴリズムを簡略化することができ、高速の符号化・復号化を行うことが可能となる。

【００３６】請求項３記載の発明において、符号化データ結合手段は、下位データのうち、差分データのビット数に相当する下位ビットを切り捨てるとともに、この下位データに差分データを結合させる。これにより、差分データと下位データとを合わせたビット数を削減することができ、高圧縮の符号化を実現できる。

【００３７】請求項４記載の発明において、データ数算出手段は、所定範囲が"０"、すなわち、値が同一の上位データを有する原データの個数を算出する。上位データが同一値であることがから、差分データのビット数は"０"となる。これにより、下位データの所定ビットを削減することなく、符号化データ列中に下位データをそのまま配列することができる。したがって、本発明によれば、原データに忠実な符号化・復号化を実現することができる。

【００３８】請求項５記載の発明において、出力手段

は、符号化データ列を構成する符号化データの個数が、原データ列を構成する原データの個数よりも少なくなった場合に符号化データ列を出力する。これにより、符号化によるデータ数の増大を回避することができる。

【００３９】請求項６記載の発明において、入力手段は、請求項１記載の符号化データ列を入力し、符号化データ分割手段は、符号化データ列を分割することにより、差分データおよび下位データを抽出する。そして、差分復号化手段は差分データに基づき上位データを生成し、復号化データ結合手段は上位データおよび下位データを結合することにより復号化データを生成する。本発明によれば、簡易なアルゴリズムで復号化処理を実現できるため、装置コストを低減できるとともに、復号化を高速に行うことが可能となる。

【００４０】請求項７記載の発明において、入力手段は、請求項２記載の符号化データ列を入力し、符号化データ分割手段は、符号化データ列を分割することにより、差分データ、下位データ、個数データを抽出する。そして、差分復号化手段は差分データに基づき上位データを生成し、復号化データ結合手段は、個数データにより表された個数分、上位データおよび下位データを結合することにより、復号化データを生成する。したがって、本発明によっても、装置コストの低減、高速の復号化を実現することが可能である。

【００４１】請求項８記載の発明において、先ず、原データ列を構成する原データを上位データおよび下位データに分割し、隣接し合う原データの上位データ間の差分データを算出する。そして、差分データが連続して所定範囲内に収まる原データの個数を算出し、個数データ、差分データ、および、下位データが配列された符号化データ列を生成する。本発明によれば、比較的に変化の少ない上位データのみの差分データを用いて符号化することにより、差分データのデータ量を低減することができる。また、上位データのみの差分を算出すればよいため、符号化・復号化アルゴリズムを簡略化することができ、高速の符号化・復号化を行うことができる。

【００４２】請求項９記載の発明において、下位データのうち、差分データのビット数に相当する下位ビットを切り捨てるとともに、この下位データに上記差分データを結合させる。これにより、差分データと下位データとを合わせたビット数を削減することができ、高圧縮の符号化を実現できる。

【００４３】請求項１０記載の発明において、複数ビットよりなる原データが複数配列された原データ列を以下の符号化モード(a)乃至(d)に従い符号化することにより、符号化データ列を生成する。

【００４４】(a)　基準となる原データとの差が所定範囲に収まる他の原データが、所定個数以上連続して配列されている場合には、上記基準となる原データと、上記連続する他の原データの個数を表す個数データとが配列された符号化データ列を生成する。この符号化モードによれば、例えば、無音状態の原データを高圧縮率で符号化することができる。

【００４５】(b)　符号化モード(a)の符号化処理を行わない場合であって、隣接し合う原データ同士の差分データが所定範囲に収まり、かつ、このような原データが連続して所定個数以上配列されている場合には、上記差分データと、上記原データの個数を表す個数データとが配列された符号化データ列を生成する。

【００４６】(c)　符号化モード(a)および(b)の符号化処理を行わない場合であって、隣接し合う原データの上位データ間の差を表す差分データが所定範囲内に収まり、かつ、このような原データが連続して所定個数以上配列されている場合には、上記原データの個数を表す個数データと、上記原データの下位データと、上記差分データとが配列された符号化データ列を生成する。

【００４７】(d)　符号化モード(a)、(b)、および、(c)の符号化処理を行わない場合には、原データが配列された符号化データ列を生成する。すなわち、原データがそのまま符号化データとして出力される。

【００４８】本発明によれば、原データに応じて最適な符号化処理を行うことができ、符号化誤差を最小限に抑えながら高圧縮率の符号化を実現することが可能となる。

【００４９】請求項１１記載の発明において、符号化処理の行われた符号化モード(a)乃至(d)のいずれかを示すデータを上記符号化データ列中に配列する。これにより、復号化時において、符号化モードに応じた復号化処理を行うことができる。

【００５０】請求項１２記載の発明において、請求項８または請求項９のいずれかに記載の符号化データ列を分割することにより、差分データおよび下位データを抽出する。そして、差分データに基づき上位データを生成し、上位データおよび下位データを結合することにより復号化データを生成する。次に、生成された複数の復号化データが配列された復号化データ列を出力する。したがって、本発明によれば、原データのうちの上位データのみを差分データに基づき復号化すれば良いため、復号化アルゴリズムを簡略化することができる。

【００５１】請求項１３記載の発明において、請求項１１記載の符号化データ列から、符号化モード(a)乃至(d)のいずれかを示すデータを抽出し、符号化モードに従い、以下の復号化処理を行う。

【００５２】(1)　抽出されたデータが符号化モード(a)である場合には、符号化データ列から、基準となる原データと、原データの個数を表す個数データとを抽出し、個数データにより示された個数の基準データが配列された復号化データ列を出力する。これにより、例えば、無音状態を表す原データを高速に復号化することができる。

【００５３】(2)　抽出されたデータが符号化モード(b)である場合には、符号化データ列から、差分データと、原データの個数を表す個数データとを抽出し、差分データに基づき復号化データを生成し、個数データにより示された個数の復号化データが配列された復号化データ列を出力する。

【００５４】(3)　抽出されたデータが符号化モード(c)である場合には、符号化データ列から、差分データと、下位データと、原データの個数を表す個数データとを抽出し、差分データに基づき上位データを生成し、上位データと下位データとを結合することにより復号化データを生成し、個数データにより示された個数の復号化データが配列された復号化データ列を出力する。

【００５５】(4)　抽出されたデータが符号化モード(d)である場合には、符号化データ列中から原データを抽出し、抽出された原データが配列された復号化データ列を出力する。

【００５６】本発明によれば、原データに応じて符号化された符号化データ列を、正しく復号化することが可能となる。また、それぞれの復号化処理を、簡易なアルゴリズムで実行することができるため、高速の復号化処理が可能となる。

【００５７】

【実施例】以下、本発明の一実施例を図面を参照しながら説明する。

【００５８】（第１実施例の構成）図１は、本発明の第１実施例に係るデータ符号化装置、データ復号化装置の概念図である。データ符号化装置１は、ワークステーション等により構成されており、ディジタルデータで表された原データ列（音声データ、画像データ等）を符号化する機能を備えたものである。データ符号化装置１において、符号化された符号化データ列はアプリケーションプログラムと一緒にＲＯＭに書き込まれ、さらにこのＲＯＭはＲＯＭカセット内に収納される。ＴＶゲーム機等のデータ復号化装置２は、ＲＯＭカセットが装着されるカートリッジを備えており、ＲＯＭカセットから読み出された符号化データ列を復号化し、復号化データ列を生成するものである。ディジタルデータである復号化データ列は、データ復号化装置２によってアナログの音声信号に変換された後、スピーカから出力される構成となっている。

【００５９】図２はデータ符号化装置１、データ復号化装置２の機能ブロック図である。データ符号化装置１は、原データ分割手段１ａ、差分算出手段１ｂ、データ数算出手段１ｃ、符号化データ結合手段１ｄ、符号化決定手段１ｅを備えて構成されている。原データ分割手段１ａには、複数の原データよりなる原データ列（ビットストリーム）が入力される構成となっている。原データ列は、アナログ音声信号を８ｂｉｔで直線量子化したディジタルデータである。なお、原データ列として、音声データに代えて、映像データ、アプリケーションプログラム等を用いても良い。

【００６０】原データ分割手段１ａは、８ｂｉｔの原データを上位４ｂｉｔの上位データ、下位４ｂｉｔの下位データに分割するものである。差分算出手段１ｂは、隣接し合う原データにおける上位データ同士の差分データを算出する機能を備えたものである。差分データは、上位データの差分を表すカウントフラグ、および、差分の符号を表す符号フラグにより構成される。例えば、第１ｂｙｔｅの上位データが"Ｆｈ"、第２ｂｙｔｅの上位データが"Ｅｈ"であるとすると、"Ｆｈ"に対する"Ｅｈ"の差は１０進表記で"－１"で表される。この"－１"を２進数で表すと、カウントフラグが"１"、符号フラグが"１"となる。また、差分データが１０進表記で"１"の場合は、カウントフラグが"１"、符号フラグが"０"となる。なお、後述する符号化モードによって、符号フラグ、あるいは、カウントフラグは使用されない場合もある。

【００６１】データ数算出手段１ｃは、連続する上位データのうち、所定範囲（±１、＋１、－１、あるいは、０）にある上位データの個数を表す個数データを算出する機能を備えている。例えば、原データ列が図３の（Ａ）に示されるように、隣接し合う７個の上位データが±１の範囲内に収まる場合には、個数データは"７ｈ"となる。なお、±１、＋１、－１、０のいずれの範囲にある上位ビットの個数が算出されるかは、後述する符号化モードにより決定される。

【００６２】符号化データ結合手段１ｄは、下位データおよび差分データを結合するとともに、図３の（Ｂ）、図４に示されるように個数データ等を予め定められたフォーマットに従い出力する機能を備えたものである。

【００６３】符号化決定手段１ｅは、予め定められた符号化モードに従い、原データ分割手段１ａ、差分算出手段１ｂ、データ数算出手段１ｃ、符号化データ結合手段１ｄを制御するものである。符号化モードは、第１～第３の３種類があり、ユーザがキーボードを操作することにより任意に変更可能である。

【００６４】データ復号化装置２は、符号化データ分割手段２ａ、差分復号化手段２ｂ、復号化データ結合手段２ｃ、復号化決定手段２ｄを備えて構成されている。符号化データ分割手段２ａは、入力された符号化データ列から、差分データ、個数データ、下位データを分離するものである。差分復号化手段２ｂは差分データ（符号フラグ、カウントフラグ）に基づき上位データを復号化する機能を備え、復号化データ結合手段２ｃは復号化された上位データ、下位データを結合する機能を備えたものである。

【００６５】例えば、符号化データ分割手段２ａは、図３の（Ｂ）に示された第３ｂｙｔｅ目の上位データのうちの上位２ｂｉｔから復号化後の下位データ"４ｈ"を

抽出し、同じく第３ｂｙｔｅ目の上位データのうちの下位２ｂｉｔから差分データ"００"を抽出する。そして、差分復号化手段２ｂは、同図（Ｂ）の第１ｂｙｔｅ目の上位データ"Ｆｈ"と、第３ｂｙｔｅ目の上位データから抽出された差分データ"００"に基づき、復号化データの第２ｂｙｔｅ目の上位データ"Ｆｈ"を算出する（同図（Ｃ））。以下、同様に復号化データの上位データ、下位データの復号化が行われる。復号化後のデータは、復号化データとしてデータ復号化データ結合手段２ｃから出力される構成となっている。

【００６６】また、復号化決定手段２ｄは、符号化決定手段１ｅによって決定された符号化モードに従い、所定の復号化方法を符号化データ分割手段２ａ、差分復号化手段２ｂ、復号化データ結合手段２ｃを制御するものである。すなわち、符号化データ分割手段２ａ、差分復号化手段２ｂ、復号化データ結合手段２ｃは、符号化モードに応じた復号化処理を行う。

【００６７】ここで、本実施例に係る符号化フォーマットを図３、図４を参照しながら説明する。図３の（Ａ）は原データ列を表し、同図の（Ｂ）は符号化データ列を表している。また、同図の（Ｃ）は復号化データ列を表している。さらに、図４は符号化データ列の第３ｂｙｔｅ以降の符号化データを表している。図３の（Ｂ）に示された符号化データ列において、第１ｂｙｔｅ目の上位４ｂｉｔのデータ"Ｆｈ"は原データ列（同図の（Ａ））の第１ｂｙｔｅ目の上位データを表している。符号化データ（同図の（Ｂ））の第１バイト目の下位４ｂｉｔのデータ"１ｈ"は、符号化データ列の先頭ｂｙｔｅを示すためのものである。すなわち、復号化装置２は、"１ｈ"の下位データを検出することにより、符号化データ列の復号化処理を実行することができる。

【００６８】符号化データ列中（同図の（Ｂ））の第２ｂｙｔｅのデータは、個数データを表している。例えば、"０７ｈ"の個数データは、７個の原データを符号化したことを表している。また、個数データは８ｂｉｔにより表現されることから、本実施例にあっては合計２５６個の原データを一度に符号化可能である。但し、原データ列が２５６個以上の原データを有する場合には、複数回の符号化により２５６個以上の原データを符号化可能である。符号化データ列の第３ｂｙｔｅ以降のデータは、原データの下位データ、および、差分データにより構成されている。

【００６９】第３ｂｙｔｅ目以降の符号化データの詳細を、図４に示す。第３ｂｙｔｅ目以降の符号化データにおいて、上位データ、下位データのそれぞれの上位２ｂｉｔのデータは、原データの下位データのうち上位２ｂｉｔのデータを表している。例えば、第１ｂｙｔｅ目の原データの下位データが"７ｈ"（２進表記で"０１１１）である場合には、"０１１１"の下位２ｂｉｔを切り捨てることにより、下位データの上位２ｂｉｔは"０１"となる。

【００７０】第３ｂｙｔｅ目以降の符号化データのうち、上位データおよび下位データのそれぞれの下位２ｂｉｔは、隣接し合う原データの上位データ間の差分データを表している。例えば、１ｂｙｔｅ目の原データの上位データ"Ｆｈ"と２ｂｙｔｅ目の原データの上位データ"Ｆｈ"との差分データは、"００"（２進表記）となる。すなわち、差分データのうちのカウントフラグおよび符号フラグはともに"０"となる。上述した下位データの上位２ｂｉｔ"０１"に、この差分データ"００"を結合することにより、"０１００"（２進表記）のデータが得られる。なお、このデータを１６進表記すると"０４ｈ"となる（図３（Ｂ）中の第３ｂｙｔｅの上位データ）。同様に、原データの第２ｂｙｔｅの下位データ"７ｈ"（図３（Ａ））のうちの上位２ｂｉｔに、原データの第２ｂｙｔｅ、第３ｂｙｔｅの上位データの差分データを結合することにより、第３ｂｙｔｅ目の符号データの下位データ"４ｈ"が求められる。

【００７１】なお、図３（Ｂ）、図４に示された符号化データ列は、第１の符号化モードにより符号化を行ったデータ列である。第１の符号化モードは、上位データ同士の差が±１の範囲にある原データを符号化するモードである。第１の符号化モード選択時においては、差分データは、差を表すカウントフラグと差の符号を表す符号フラグとの合計２ｂｉｔより構成される。この場合、第３ｂｙｔｅ以降の符号化データにおいては、差分データおよび下位データを併せて４ｂｉｔで表さなければならない。したがって、原データ列中の４ｂｉｔの下位データの下位２ｂｉｔを切り捨てなければならず、この切り捨て誤差は復号化時に雑音となって現れる。

【００７２】第２の符号化モードは、上位データ同士の差が＋１若しくは－１の原データを符号化するモードである。第１の符号化モード選択時においては、差分データの符号は正または負のいずれか一方であるため、差分データのうちの符号フラグは存在しない。したがって、差分データは差分を表すカウントフラグの１ｂｉｔのみとなり、原データの下位データを３ｂｉｔで表すことができる。このため、原データにおける下位データの切り捨て誤差は１ｂｉｔのみである。

【００７３】第３の符号化モードは、上位データが同一の原データを符号化するモードである。第３の符号化モード選択時においては、差分データは存在しないため、原データの下位データを４ｂｉｔで表すことができる。したがって、原データにおける下位データの切り捨て誤差は０ｂｉｔとなり、符号化誤差のない符号化処理を実現できる。

【００７４】なお、第１、第２の符号化モードにおいて、下位データのうち、切り捨てられたｂｉｔの平均値を復号化時に加算することにより、符号化誤差を低減することが可能である。

【００７５】（第１実施例の作用）続いて、本実施例に係るデータ符号化装置、データ復号化装置の作用を説明する。

【００７６】図６は、データ符号化装置の作用を表すメインフローチャートである。先ず、データ符号化装置１は、原データの入力が終了したか否かを判断し（Ｓ１０１）、入力が終了した場合（Ｓ１０１でＹＥＳ）には符号化処理を終了する（Ｓ１０２）。一方、原データの入力が終了していない場合（Ｓ１０１でＮＯ）には、データ符号化装置１は入力された原データを順に読み込む（Ｓ１０３）。そして、符号化決定手段１ｅは、差が±１の上位データが６ｂｙｔｅ以上連続したか否かを判断する（Ｓ１０４）。例えば、図３の（Ａ）に示される原データのように、差が±１の上位データが６ｂｙｔｅ以上連続する場合には、データ符号化装置１はデータ符号化の処理を実行し（Ｓ１０５）、Ｓ１０１の処理に戻る。これにより、原データに対する符号化が行われ、符号化装置１から符号化デー列が出力される。

【００７７】なお、Ｓ１０４の判断において、差が±１の上位データが６ｂｙｔｅ以上連続するか否かを判断することにより、符号化データのｂｙｔｅ数が原データのｂｙｔｅ数を上回るのを防止することができる。すなわち、原データ列が５ｂｙｔｅ以下である場合には、符号化データ列のｂｙｔｅ数は原データ列のｂｙｔｅ数以上となり、データ圧縮の効果を望めない。そこで、本実施例にあっては、図５に示されるように、原データ列が６ｂｙｔｅ以上である場合に限り、符号化処理を実行することとしている。

【００７８】差が±１の上位データが６ｂｙｔｅ以上連続しない場合（Ｓ１０４でＮＯ）には、符号化決定手段１ｅは、入力された原データの中に下位データ"１ｈ"があるか否かを判断する（Ｓ１０６）。"１ｈ"の下位データがある場合（Ｓ１０６ｄｅＹＥＳ）には、符号化決定手段１ｅは"１ｈ"の下位データを"０ｈ"に変更する（Ｓ１０７）。上述したように、"１ｈ"の下位データは符号化データ列の先頭ｂｙｔｅを表すものであるため、符号化を行わない場合（Ｓ１０４でＮＯ）には、復号化装置２によって誤って復号化処理がなされるのを防止する必要がある。このため、符号化を行わない場合には、"１ｈ"の下位データを"０ｈ"に書き換えることとしている。Ｓ１０６の判断の結果、"１ｈ"の下位データがない場合には、符号化決定手段１ｅは原データを変更することなく、そのまま原データを出力する。この後、符号化装置１は、Ｓ１０１以降の処理を繰り返し実行する。以上の処理により、例えば音声等を表す原データ列の符号化が行われ、データ容量の少ない符号化データ列が生成される。

【００７９】続いて、本実施例に係るデータ復号化処理を説明する。

【００８０】図７は、本実施例に係るデータ復号化処理のメインルーチンを表すフローチャートである。この図において、データ復号化装置２は、入力された符号化データに対する復号化処理が終了したか否かを判断する（Ｓ３０１）。復号化処理が終了したとデータ復号化装置２が判断すると（Ｓ３０１でＹＥＳ）、データ復号化装置２は全ての処理を終了する（Ｓ３０２）。一方、入力された符号化データに対する復号化処理が終了していないとデータ復号化装置２が判断した場合（Ｓ３０１でＮＯ）には、データ復号化装置２はＳ３０３以降の処理を実行する。

【００８１】Ｓ３０３において、データ復号化装置２は入力されたデータ列を読み込み、復号化決定手段２ｄは、読み込んだデータ列が符号化データ列か否かを判断する（Ｓ３０４）。すなわち、復号化決定手段２ｄは、先頭ｂｙｔｅの下位データが符号化データ列の先頭を表す"１ｈ"であるかどうかを判断する。入力されたデータ列が符号化データ列でないとデータ復号化装置２が判断した場合（Ｓ３０４でＮＯ）には、入力されたデータ列をそのまま出力し（Ｓ３０６）、Ｓ３０１に戻る。一方、入力されたデータ列が符号化データ列であるとデータ復号化装置２が判断した場合（Ｓ３０４でＹＥＳ）には、符号化データ分割手段２ａ等は符号化データ列の復号化を実行する（Ｓ３０５）。このようにして、入力された符号化データ列は順に復号化され（Ｓ３０４でＹＥＳ）、データ復号化装置２から出力される。

【００８２】図８は、Ｓ３０５の復号化処理のサブルーチンを表すフローチャートである。Ｓ４０１において、先ず符号化データ分割手段２ａは、符号化データ列の第１ｂｙｔｅの下位データをクリアし（Ｓ４０１）、この第１ｂｙｔｅのデータを、第１ｂｙｔｅの復号化データの上位データとして保存する。例えば、符号化データ分割手段２ａは、図３の（Ｂ）に示されるように、符号化データ列の第１ｂｙｔｅの下位データ"１ｈ"をクリアし、第１ｂｙｔｅの上位データ"Ｆｈ"を保存する。

【００８３】そして、符号化データ分割手段２ａは、符号化データ列の第２ｂｙｔｅの個数データを抽出し、この個数データの値を本サブルーチンにおけるカウンタＮの最大値として保存する（Ｓ４０３）。続いて、符号化データ分割手段２ａは、カウンタＮの値をインクリメントする（Ｓ４０４）。なお、カウンタＮの初期値は"０"であるため、インクリメント後のカウンタＮの値は"１"となる。そして、符号化データ分割手段２ａは、カウンタＮの値が最大値を超えたか否か、すなわち、カウンタＮの値が個数データの値を超えたか否かを判断する（Ｓ４０５）。上述したように、個数データは少なくとも"６"以上であり、また、このときのカウンタの値は"１"である。したがって、カウンタＮの値は個数データの値よりも小さくなり（Ｓ４０５でＮＯ）、Ｓ４０６以降の処理が実行される。

【００８４】Ｓ４０６において、符号化データ分割手段

２ａは、符号化データ中の（Ｎ＋４）番目の４ｂｉｔデータを読む。このときのＮの値は"１"であるため、符号化データ分割手段２ａは、符号化データ中の４ｂｉｔのデータの５番目、すなわち符号化データ中の第３ｂｙｔｅ目の上位４ｂｉｔのデータを読む。例えば、図３（Ｂ）において、符号化データ列の第３ｂｙｔｅ目の上位４ｂｉｔのデータ"４ｈ"が読み出される。

【００８５】次に、符号化データ分割手段２ａは、読み出された４ｂｉｔデータ"４ｈ"から差分データ、下位データを分離する（Ｓ４０７）。すなわち、符号化データ分割手段２ａは読み出された４ｂｉｔデータのうち、上位２ｂｉｔを下位データとして抽出し、下位２ｂｉｔを差分データとして抽出する。符号化データ列から読み出された４ｂｉｔデータ"４ｈ"（図３（Ｂ））を２進数で表すと、"０１００"となる。したがって、このデータの上位２ｂｉｔ"０１"を抽出し、抽出された"０１"のデータを左に２ｂｉｔシフトすることにより、復号化後の下位データ"０１００"（１６進表記で"４ｈ"）が得られる。また、読み出されたデータ"０１００"の下位２ｂｉｔを抽出することにより、差分データ"００"が求められる。さらに、符号化データ分割手段２ａは、この差分データをカウントフラグ"０"と符号フラグ"０"とに分離する。

【００８６】差分復号化手段２ｂは、第Ｎｂｙｔｅ目の復号化データにおける上位データに対して、差分データを加減し、復号化データの第（Ｎ＋１）ｂｙｔｅ目の上位データを復号化する（Ｓ４０８）。すなわち、差分復号化手段２ｂは、第１ｂｙｔｅ目の復号化データにおける上位データ"Ｆｈ"に対して、差分データを加減することにより、復号化データの第２ｂｙｔｅ目の上位データを復号化する。このときの差分データは"００"であるため、復号化データの第２ｂｙｔｅの上位データは、復号化データの第１ｂｙｔｅの上位データ"Ｆｈ"に等しくなる（図３（Ｃ））。

【００８７】そして、復号化データ結合手段２ｃは、復号化データのうちの第１ｂｙｔｅの下位データ"４ｈ"と、第１ｂｙｔｅの上位データ"Ｆｈ"（Ｓ４０２において算出済み）とを結合する（Ｓ４０９）。すなわち、復号化データ結合手段２ｃは、上位データ"Ｆｈ"を左に４ｂｉｔシフトすることにより"Ｆ０ｈ"を求め、この"Ｆ０ｈ"と下位データ"４ｈ"との論理和を算出する。これにより、第１ｂｙｔｅ目の復号化データ（原データ）"Ｆ４"が求められる。

【００８８】この後、復号化装置２はＳ４０４に戻り、Ｓ４０４～Ｓ４０９の処理を繰り返し実行する。Ｓ４０４において、符号化データ分割手段２ｄは、カウンタＮの値をインクリメントすることにより、カウンタＮの値を"２"にする。そして、符号化データ分割手段２ａは、カウンタＮの値が最大値を超えたか否かを判断し（Ｓ４０５）、判断の結果がＮＯであればＳ４０６以降の処理を実行する。

【００８９】Ｓ４０６において、符号化データ分割手段２ａは、符号化データ中の（Ｎ＋４）番目の４ｂｉｔデータを読む。すなわち、符号化データ分割手段２ａは、符号化データ列を４ｂｉｔ毎に区切り、符号化データ列の先頭から（Ｎ＋４）番目の４ｂｉｔデータを読む。このときのＮの値は"２"であるため、符号化データ分割手段２ａは、符号化データ中の４ｂｉｔのデータの６番目、すなわち符号化データ中の第３ｂｙｔｅの下位４ｂｉｔのデータを読む。例えば、図３（Ｂ）において、符号化データ列の第３ｂｙｔｅの下位４ｂｉｔのデータ"４ｈ"が読み出される。

【００９０】次に、符号化データ分割手段２ａは、読み出された４ｂｉｔデータ"４ｈ"から差分データ、下位データを分離する（Ｓ４０７）。分離された差分データ、下位データの値は、それぞれ"００"（２進表記）、"４ｈ"となる。差分復号化手段２ｂは、第２ｂｙｔｅ目の復号化データにおける上位データ"Ｆｈ"に対して、差分データを加減することにより、復号化データの第３ｂｙｔｅ目の上位データ"Ｆｈ"を復号化する（Ｓ４０８、図３（Ｃ））。なお、第２ｂｙｔｅ目の復号化データにおける上位データ"Ｆｈ"は、前回実行されたＳ４０８の処理によって算出されたものである。この後、復号化データ結合手段２ｃは、復号化データのうちの第２ｂｙｔｅの下位データ"４ｈ"と、第２ｂｙｔｅの上位データ"Ｆｈ"とを結合することにより、第２ｂｙｔｅの復号化データ"Ｆ４"を算出する（Ｓ４０９）。

【００９１】以上の処理を繰り返すことにより、図３（Ｂ）に示された符号化データが順に復号化される。そして、全ての符号化データの復号化が終了すると、カウンタＮの値が最大値を超えるため（Ｓ４０５でＹＥＳ）、データ復号化装置２の処理は図７のメインフローチャートに戻る。

【００９２】以上説明した処理により、音声データ等の符号化および復号化符号化を行うことができる。一般に、音声データ、画像データ等は緩やかに変化することが多いため、上位データの変化量は比較的に少ない。したがって、本実施例において示されたように、原データのうちの上位データのみを差分符号化するとともに、差分符号化した上位データの個数を個数データとして表すことにより、音声データ等を高圧縮率で符号化・復号化することができる。また、データ符号化およびデータ復号化に要する演算は差分演算、データ並べ換え等の処理であるため、簡易なアルゴリズムで、高圧縮率の符号化・復号化を実現できる。

【００９３】（第２実施例）続いて、本発明の第２実施例に係るデータ符号化装置、データ復号化装置を説明する。本実施例に係るデータ符号化装置、データ復号化装置は、図１および図２に示された第１実施例に係る構成

と略同様である。しかし、本実施例に係る符号化決定手段１ｅ（図２）は、原データ列に応じて最適な符号化モードを自動的に選択可能である点において、第１実施例と異なっている。

【００９４】本実施例に係る符号化データ列のフォーマットを図９に示す。この図において、第１ｂｙｔｅの上位４ｂｉｔのデータは符号化モードを表している。例えば、第１ｂｙｔｅの上位４ｂｉｔのデータ"１ｈ"は、第２の符号化モードにより符号化を行ったことを示している。なお、本実施例に係る符号化モードは、後述するように第１～第８の８種類ある。さらに、同図の第１ｂｙｔｅの下位４ｂｉｔのデータ、第２ｂｙｔｅの８ｂｉｔのデータは、符号化を行った原データの個数を表す個数データである。個数データは合計１２ｂｉｔにより表されるため、最大４０９５個までの原データ列を同一の符号化モードで符号化することができる。第３ｂｙｔｅ以降のデータは符号化データを表しており、これらのデータは符号化モードに対応した方法により符号化されたものである。

【００９５】図１１に、本実施例に係る符号化モードの一覧を示す。以下に、これらの符号化モードを順に説明する。

【００９６】第１の符号化モード（符号化モード(c)）は、第１実施例に係る第３の符号化モードに対応するものである。すなわち、この第１の符号化モードは、同一の上位データを有する原データを符号化するモードである。第１の符号化モード選択時においては、差分データは存在しないため、原データの下位データを４ｂｉｔで表すことができる。したがって、原データにおける下位データの切り捨て誤差は０ｂｉｔとなる。

【００９７】第２の符号化モード（符号化モード(c)）は、第１実施例に係る第２の符号化モードに対応するものである。この第２の符号化モードは、上位データ同士の差が＋１である原データを符号化するモードである。第２の符号化モード選択時においては、差分データの符号は正のみであるため、差分データのうちの符号フラグは存在しない。したがって、差分データは差分を表すカウントフラグの１ｂｉｔのみとなり、原データの下位データを３ｂｉｔで表すことができる。このため、原データにおける下位データの切り捨て誤差は１ｂｉｔとなる。

【００９８】第３の符号化モード（符号化モード(c)）は、第１実施例に係る第２の符号化モードに対応するものであり、上位データ同士の差が－１である原データを符号化するモードである。第３の符号化モードにおける他の符号化方法については、第２の符号化モードと同様である。

【００９９】第４の符号化モード（符号化モード(c)）は、第１実施例に係る第１の符号化モードに対応するものである。すなわち、この第４の符号化モードは、上位データ同士の差が±１の範囲にある原データを符号化するモードである。第４の符号化モード選択時においては、差分データは、差を表すカウントフラグと差の符号を表す符号フラグとの合計２ｂｉｔより構成されるため、原データ列中の下位データを２ｂｉｔで表現しなければならない。このため、４ｂｉｔの下位データのうちの下位２ｂｉｔは、切り捨て誤差として復号化時に雑音となって現れる。

【０１００】ここで、第１～第４の符号化モード選択時における符号化データ列を図１１の（Ａ）に示す。符号化データ列中の第１ｂｙｔｅの上位４ｂｉｔのデータは、符号化モードを表している。例えば、このデータが"３ｈ"である場合には、第４の符号化モードに従い符号化が行われたことを復号化時に判断することができる。符号化を行った原データの個数を表す個数データは、第１ｂｙｔｅの下位の４ｂｉｔのデータ、第２ｂｙｔｅの８ｂｉｔのデータの合計１２ｂｉｔに書き込まれる。また、原データの下位データは、差分データと併せて符号化データ列の第３ｂｙｔｅ以降に格納される。なお、本実施例に係る第１～第４の符号化モードは、第１実施例に係る符号化モードと異なり、原データの第１ｂｙｔｅ目の上位データは符号化データ列中に格納されない。したがって、第１～第４符号化モードによる符号化データ列を復号化する場合には、直前に復号化された復号化データの上位データを基準として、符号化データ列中の上位データが順に復号化される。

【０１０１】続いて、第５、第６の符号化モード（符号化モード(b)）について説明する。これらの符号化モードは、いわゆる差分符号化を行うモードである。原データと次の原データとの差が＋２に収まり、かつ、このような原データが４ｂｙｔｅ以上連続する場合に、第５の符号化モードが採用される。また、第６の符号化モードは、原データと次の原データとの差が－２に収まり、かつ、このような原データが４ｂｙｔｅ以上連続する場合に、採用される。第５、第６の符号化モード選択時における符号化データ列を図１１（Ｂ）に示す。この図に示されるように、第３ｂｙｔｅ目以降のの符号化データは、隣接し合う原データ間の差分データを表している。なお、第１ｂｙｔｅ目の原データについての差分データは、第１ｂｙｔｅ目の原データと、第１ｂｙｔｅ目の原データの直前の原データとの差に基づき算出される。

【０１０２】第５、第６の符号化モードによれば、差分データは２ｂｉｔにより表されるため、４ｂｙｔｅ分の原データを１ｂｙｔｅの符号化データに圧縮することができる。但し、符号化モードを表すデータ、個数データを符号化データ列中に設ける必要があるため、４個未満の原データを符号化した場合には、符号化データ列のデータ量が原データのデータ量よりも増えてしまう。そこで、差が±２以内の原データが４個以上連続する場合に限り、第５、第６の符号化モードによる符号化を行って

いる。

【０１０３】第７の符号化モード（符号化モード(d)）は、原データをそのまま符号化データ列中に格納するモードである。但し、符号化データ列の第１、第２ｂｙｔｅには符号化モードを表すデータ、個数データが格納されるため、全ての原データを符号化データ列中に格納したのでは、符号化によりデータ量が増加してしまう。そこで、図１２の（Ａ）に示されるように、原データの第１～第４ｂｙｔｅについては上位データのみを符号化データ列中に格納することにより、データ量の増加を回避している。第５ｂｙｔｅ以降の原データは、符号化データ列中の第５ｂｙｔｅ以降にそのまま格納される。したがって、第７の符号化モードによれば、原データのデータ量に対する、符号化データのデータ量の比率（圧縮率）は１００％となる。

【０１０４】第８の符号化モード（符号化モード(a)）は、基準となる原データ（ｄａｔａ１）に対する差が±ｄの範囲内の原データを符号化するモードである。第８の符号化モード選択時における符号化データ列を図１２の（Ｂ）に示す。この図において、符号化データの第３ｂｙｔｅには、第１ｂｙｔｅの原データ格納される。そして、符号化データ列のうちの第１ｂｙｔｅの下位データ、および、第２ｂｙｔｅには、第１ｂｙｔｅの原データとの差が±ｄの範囲内にあり、かつ、連続した原データの個数を表すデータが格納される。また、１つの符号化データ列は３ｂｙｔｅの符号化データから構成される。このため、±ｄの範囲内の原データが３個以上連続する場合に、第８の符号化モードが選択される。

【０１０５】このような第８の符号化モードによれば、原データのうちの変化の少ない部分を高圧縮率で符号化・復号化することが可能である。例えば、微少信号の原データ（音声データ）が連続するような場合には、これらの原データの再生音は聴感上、無音として捉えることができる。したがって、このような場合には、所定範囲内（ｄの範囲内）の原データの個数を符号化データ列に格納することにより、極めて高圧縮の符号化・復号化を実現することができる。なお、ｄの値はユーザ等によって任意に設定可能である。例えば、原データが音声データであるような場合には、無音に近い値（例えば、ｄ＜８）にｄを設定するのが望ましい。

【０１０６】続いて、本実施例の作用を図１３に示されたフローチャートを参照しながら説明する。このフローチャートは、符号化処理を表すメインフローチャートである。本実施例に係るデータ符号化装置１（図２）は、原データに応じて上述した８種類の符号化モードのいずれかを自動的に選択し、選択した符号化モードに基づき原データを符号化することが可能である。図１３に示されるように、符号化モードの選択可否の判断は、圧縮率の高い符号化モードから順になされる。

【０１０７】先ず、符号化決定手段１ｅは、基準となる原データ（図１２（Ｂ）中のｄａｔａ１）との差が±ｄの範囲内にあり、かつ、このような原データが連続して３ｂｙｔｅ以上連続しているか否かを判断する（Ｓ５０１）。判断の結果がＹＥＳであれば、符号化決定手段１ａは第８の符号化モードを選択する。そして、原データ分割手段１ａ、差分検出手段１ｂ、個数データ算出手段１ｃ、符号化データ結合手段１ｄは、第８の符号化モードに従い原データの符号化を行う（Ｓ５０２）。例えば、原データのうち無音状態に相当する部分が、この第８の符号化モードにより符号化される（図１２（Ｂ））。

【０１０８】一方、Ｓ５０１の判断の結果がＮＯである場合には、符号化決定手段１ｅは、さらにＳ５０３の判断を実行する。Ｓ５０３において、符号化決定手段１ｅは、下位データが±２以内に収まる原データが４個以上連続するか否かを判断する。判断の結果がＹＥＳであれば、符号化決定手段１ｅは第５、第６の符号化モードを選択し、原データ分割手段１ａ等は第５、第６の符号化モードに従い原データの符号化を実行する（Ｓ５０４）。これにより、原データに対していわゆる差分ＰＣＭによる符号化が行われる（図１１（Ｂ））。

【０１０９】Ｓ５０３の判断の結果がＮＯである場合には、符号化決定手段１ｅはＳ５０５の判断を実行する。すなわち、符号化決定手段１ｅは、上位データが±１に収まる原データが６個以上連続するか否かを判断する（Ｓ５０５）。そして、判断の結果がＹＥＳであれば、符号化決定手段１ｅは第４の符号化モードを選択する。原データ分割手段１ａ等は第４の符号化モードに従い、符号化を実行する（Ｓ５０６）。すなわち、差分算出手段１ｂは、原データの上位データ同士の差分データを２ｂｉｔで表すとともに、原データの下位データの２ｂｉｔを切り捨てる。そして、符号化データ結合手段１ｄは、下位データおよび差分データを、符号化データ列中の第３ｂｙｔｅ以降に格納する（図１１（Ａ））。

【０１１０】Ｓ５０５の判断の結果がＮＯである場合には、符号化決定手段１ｅはＳ５０７の判断を実行する。Ｓ５０７において、符号化決定手段１ｅは、上位データが－１以内に収まる原データが６個以上連続するか否かを判断する。この判断の結果がＹＥＳである場合には、符号化決定手段１ｅは第３の符号化モードを選択する。原データ分割手段１ａ等は第３の符号化モードに従い、原データの符号化を実行する（Ｓ５０８）。すなわち、差分算出手段１ｂは、原データの上位データ同士の差分データを１ｂｉｔで表すとともに、原データの下位データのうちの下位１ｂｉｔを切り捨てる。そして、符号化データ結合手段１ｄは、下位データおよび差分データを、符号化データ列中の第３ｙｔｅ以降に格納する（図１１（Ａ））。

【０１１１】Ｓ５０７の判断の結果がＮＯである場合には、符号化決定手段１ｅはＳ５０９の判断を実行する。

Ｓ５０９において、符号化決定手段１ｅは、上位データが＋１以内に収まる原データが６個以上連続するか否かを判断する。判断の結果がＹＥＳである場合には、符号化決定手段１ｅは第２の符号化モードを選択し、原データ分割手段１ａ等は第２の符号化モードに従い原データの符号化を実行する（Ｓ５１０）。第２の符号化モードは、上述した第３の符号化モードに係る差分データを正の符号で表した符号化モードである（図１１（Ａ））。

【０１１２】Ｓ５０９の判断の結果がＮＯである場合には、符号化決定手段１ｅはＳ５１１の判断を実行する。Ｓ５１１において、符号化決定手段１ｅは、同一の上位データが６個以上連続するか否かを判断する。そして、判断の結果がＹＥＳである場合には、符号化決定手段１ｅは第１の符号化モードに従い原データの符号化を実行する（Ｓ５１２、図１１（Ａ））。第１の符号化モードによれば、上位データ同士の差を表す差分データは存在しないため、原データの下位データをそのまま符号化データ列中に格納することができる。したがって、下位データの切り捨て誤差は０ｂｉｔとなり、復号化時における雑音の発生等を回避することが可能となる。

【０１１３】最後に、Ｓ５１１の判断の結果がＮＯである場合には、符号化決定手段１ｅは、第７の符号化モードを選択する。原データ分割手段１ａ等は、図１２（Ａ）に示された符号化フォーマットに従い、原データを符号化データ列中に順に格納する（図１２（Ａ））。この第７の符号化モードによれば、第１～第４の原データを除き、原データはそのまま符号化データ列中に格納される。したがって、第７の符号化モードが選択された場合には、データ圧縮の効果は得られない。

【０１１４】以上の処理によって、原データ列の一部の符号化が終了する。そして、データ符号化装置１の処理は、図示されていないメインフローチャートに戻った後、再度このフローチャートを実行する。このようにして、原データ列の部分毎に最適な符号化モードにより符号化が行われる。データ復号化装置２は、符号化モードを表すデータを符号化データ列から順次検出し、符号化モードに応じた復号化処理を実行する。したがって、本実施例によれば、音質・画質の劣化を最小限に抑えながら高圧縮の符号化・復号化を実現できる。これにより、ＴＶゲーム機等においては、データ容量の限られたＲＯＭカセット等に、より多くの音声データ、画像データ等を格納することが可能となる。

【０１１５】なお、本発明は、上述した実施例に限定されることなく、本発明の趣旨を逸脱しない範囲において実施可能である。例えば、８ｂｉｔの原データに限らず、１６ｂｉｔ、３２ｂｉｔ等の原データについて、上述した実施例を適用してもよい。また、原データの上位データ、下位データを必ずしも４ｂｉｔ毎に分割しなくても良い。例えば上位データに２ｂｉｔ、下位データに６ｂｉｔを割り当てることも可能である。

【０１１６】

【発明の効果】以上説明してきたように、本発明によれば、原データのうち、比較的に変化の少ない上位データのみの差分データを用いて符号化することにより、通常の差分ＰＣＭに比べて差分データのデータ量を低減することができる。すなわち、ＴＶゲーム機等のマルチメディアにおいて使用される音声データ、画像データは、一般に上位データの変化量が比較的に少ない。このため、上位データの差分データのデータ量を低減することができ、高圧縮の符号化・復号化を実現することが可能となる。さらに、本発明にあっては、比較的に変化の少ない上位データの差分を算出するため、差分データが予め定められたデータ長を超えることはなく、これにより符号化・復号化誤差を低減することができる。

【０１１７】また、上位データのみの差分を算出すればよいため、符号化・復号化アルゴリズムを簡略化することができる。したがって、例えば、家庭用ＴＶゲーム機等においては、装置の製造コストを低減することが可能となる。さらに、簡易なアルゴリズムで符号化・復号化を実現できるため、高速の符号化・復号化を行うことができる。

【図面の簡単な説明】

【図１】本発明の第１実施例に係るデータ符号化装置、データ復号化装置の概念図である。

【図２】本発明の第１実施例に係るデータ符号化装置、データ復号化装置の機能ブロック図である。

【図３】本発明の第１実施例に係る原データ列、符号化データ列、復号化データ列の一例を示す図である。

【図４】本発明の第１実施例に係る符号化データのフォーマットを説明するための図である。

【図５】本発明の第１実施例に係る符号化データのフォーマットを説明するための図である。

【図６】本発明の第１実施例に係るデータ符号化方法を表すフローチャートである。

【図７】本発明の第１実施例に係るデータ復号化方法を表すフローチャートである。

【図８】本発明の第１実施例に係る復号化処理の詳細を表すフローチャートである。

【図９】本発明の第２実施例に係る符号化データ列の一例を表す図である。

【図１０】本発明の第２実施例に係る符号化モードを表す図である。

【図１１】本発明の第２実施例に係る、第１～第４の符号化モード、第５、第６の符号化モードの符号化フォーマットを表す図である。

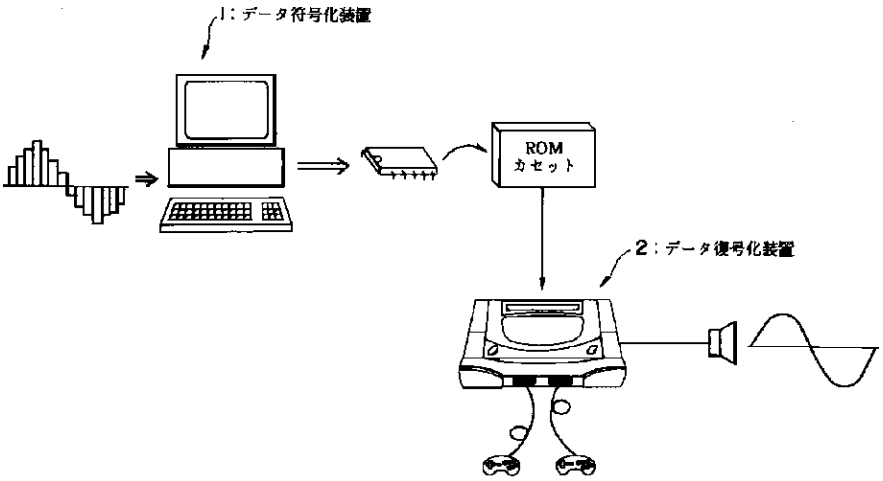
【図１２】本発明の第２実施例に係る、第７の符号化モード、第８の符号化モードの符号化フォーマットを表す図である。

【図１３】本発明の第２実施例に係るデータ符号化方法を表すフローチャートである。

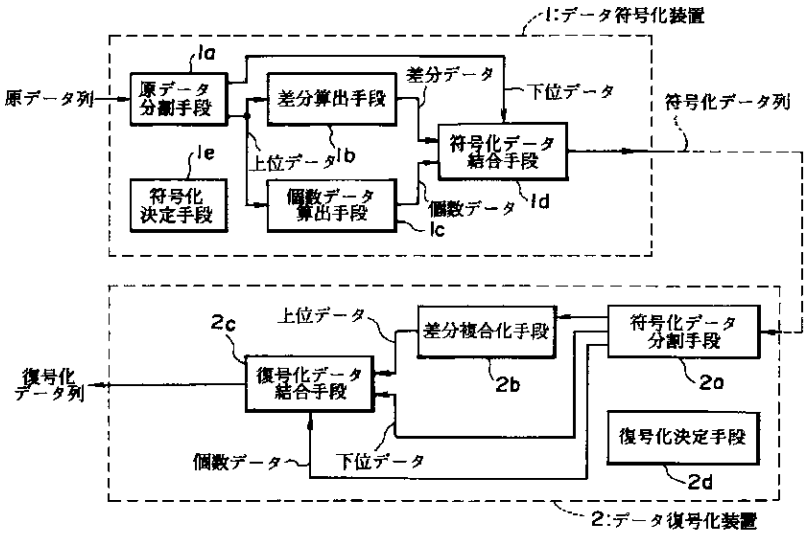
【符号の説明】

１　データ符号化装置
１ａ　原データ分割手段
１ｂ　差分算出手段
１ｃ　個数データ算出手段
１ｄ　符号化データ結合手段
＊１ｅ　符号化決定手段
２　データ復号化装置
２ａ　符号化データ分割手段
２ｂ　差分復号化装置
２ｃ　復号化データ結合手段
＊

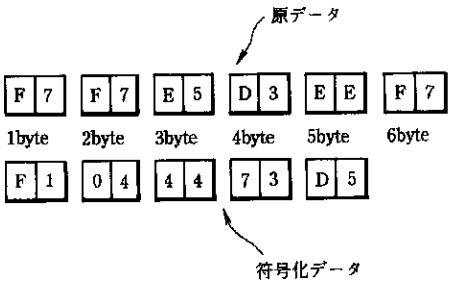
【図１】



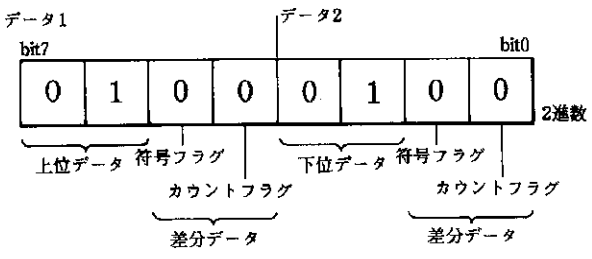
【図２】



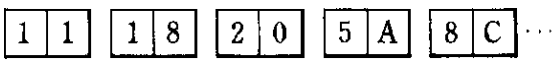
【図５】



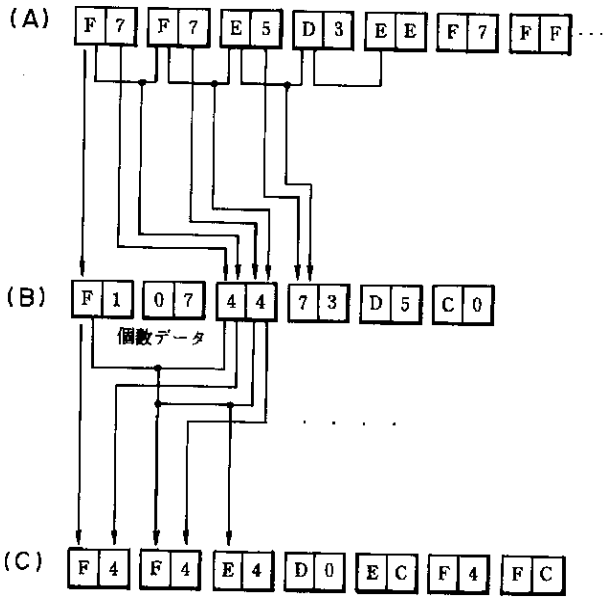
【図４】



【図９】



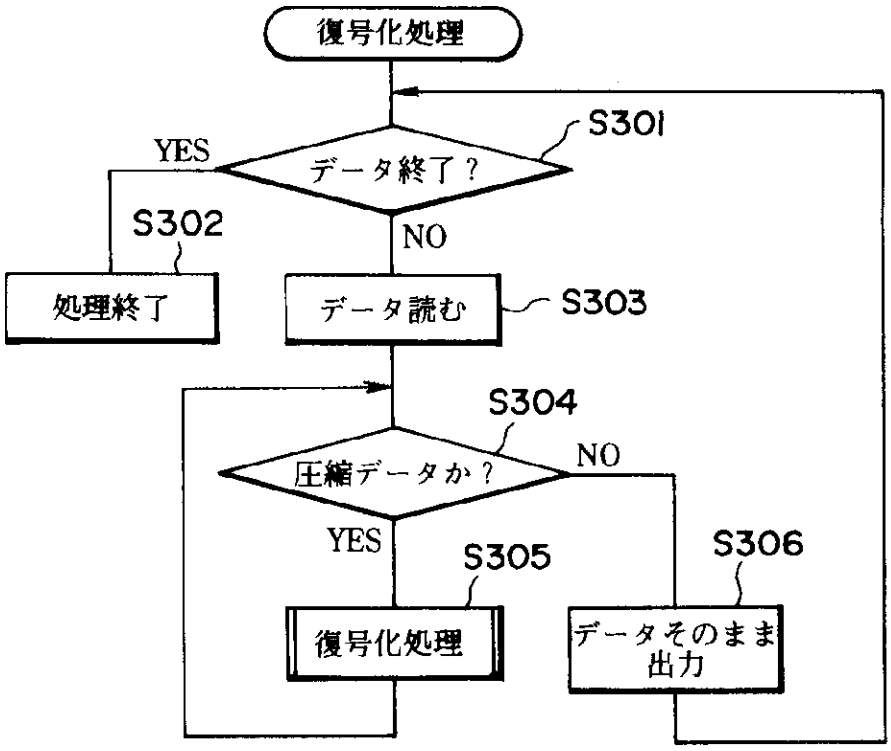
【図３】



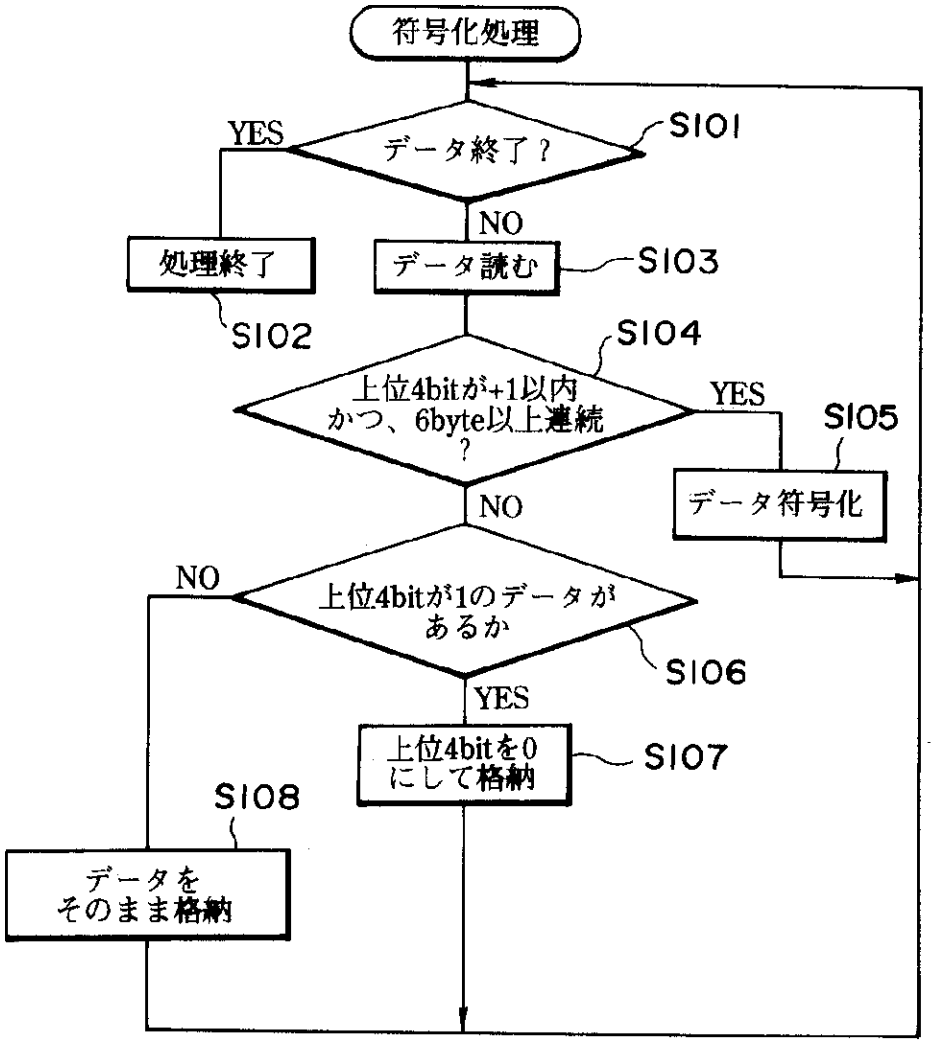
【図１０】

1	切り捨てbit数0の方式
2	切り捨てbit数1の方式(切り捨てbitを加算に使用)
3	切り捨てbit数1の方式(切り捨てbitを加算に使用)
4	切り捨てbit数2の方式
5	前データとの差を2bitで圧縮する方式(差分PCM) 但し、この2bitデータは加算に使用。
6	前データとの差を2bitで圧縮する方式(差分PCM) 但し、この2bitデータは減算に使用。
7	1byteモード。 原データを直接持つ方式。
8	振幅0モード。 振幅0 (無音状態) を表現する方式。

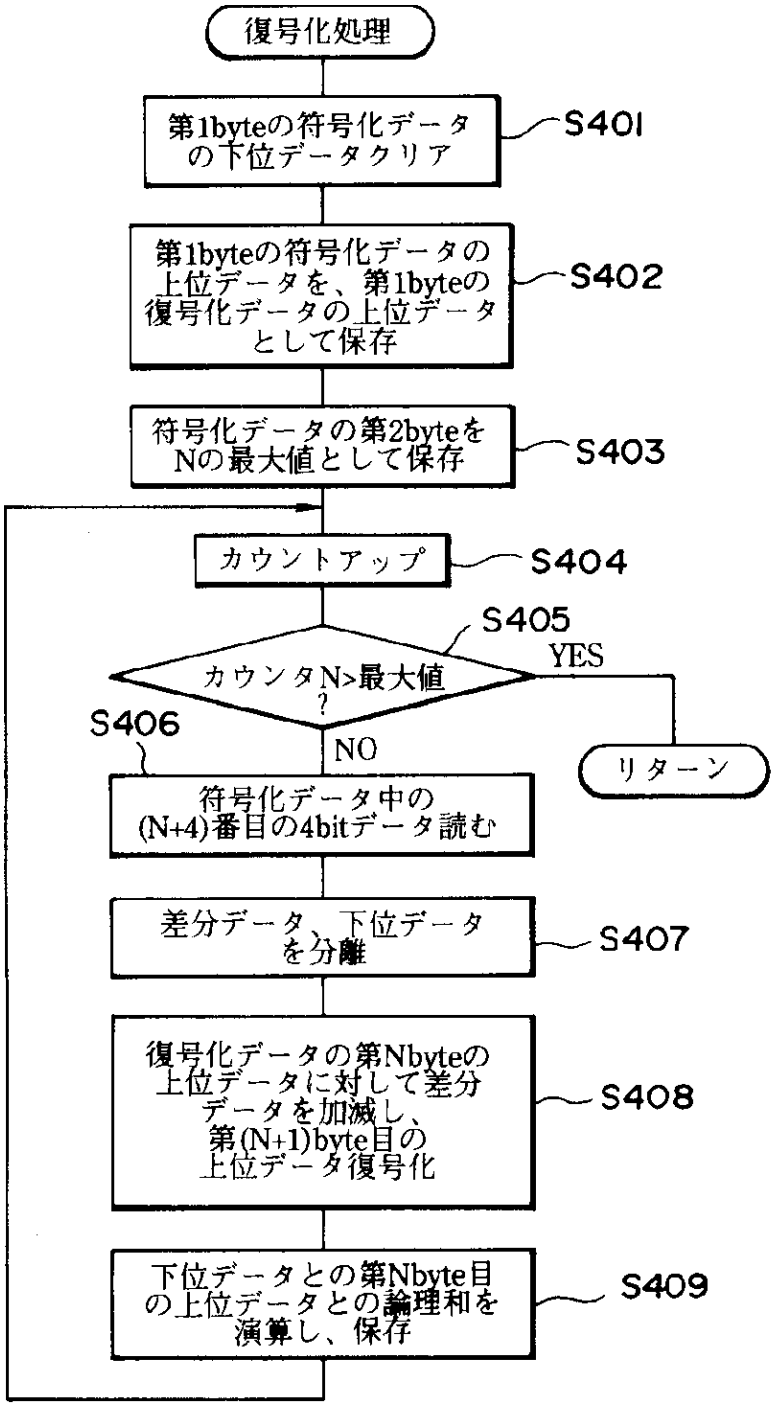
【図７】



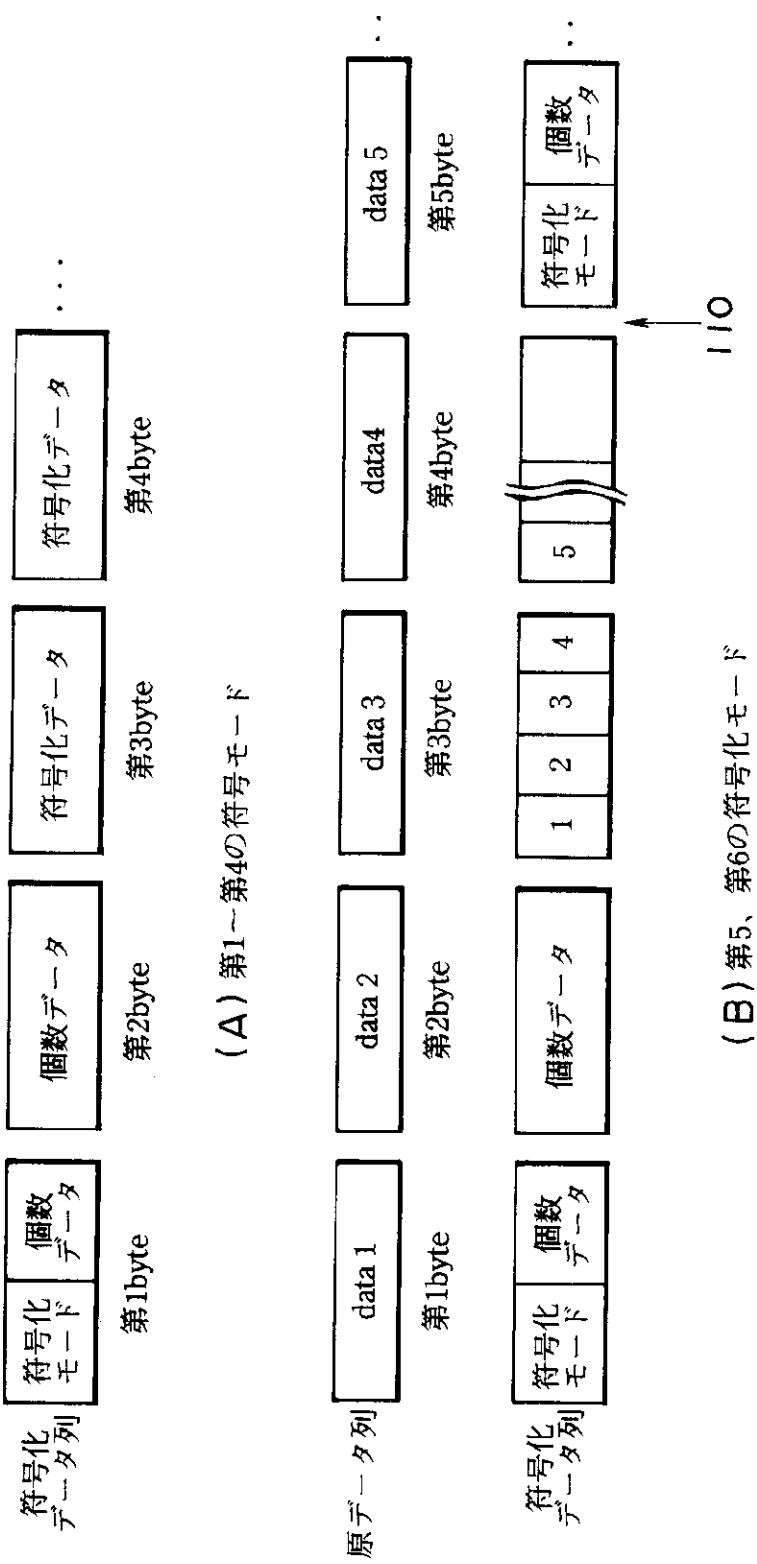
【図６】


符号化処理
データ終了？
S101
YES
NO
処理終了
S102
データ読む
S103
S104
上位4bitが+1以内
かつ、6byte以上連続
？
YES
S105
データ符号化
NO
NO
上位4bitが1のデータが
あるか
S106
YES
上位4bitを0
にして格納
S107
S108
データを
そのまま格納

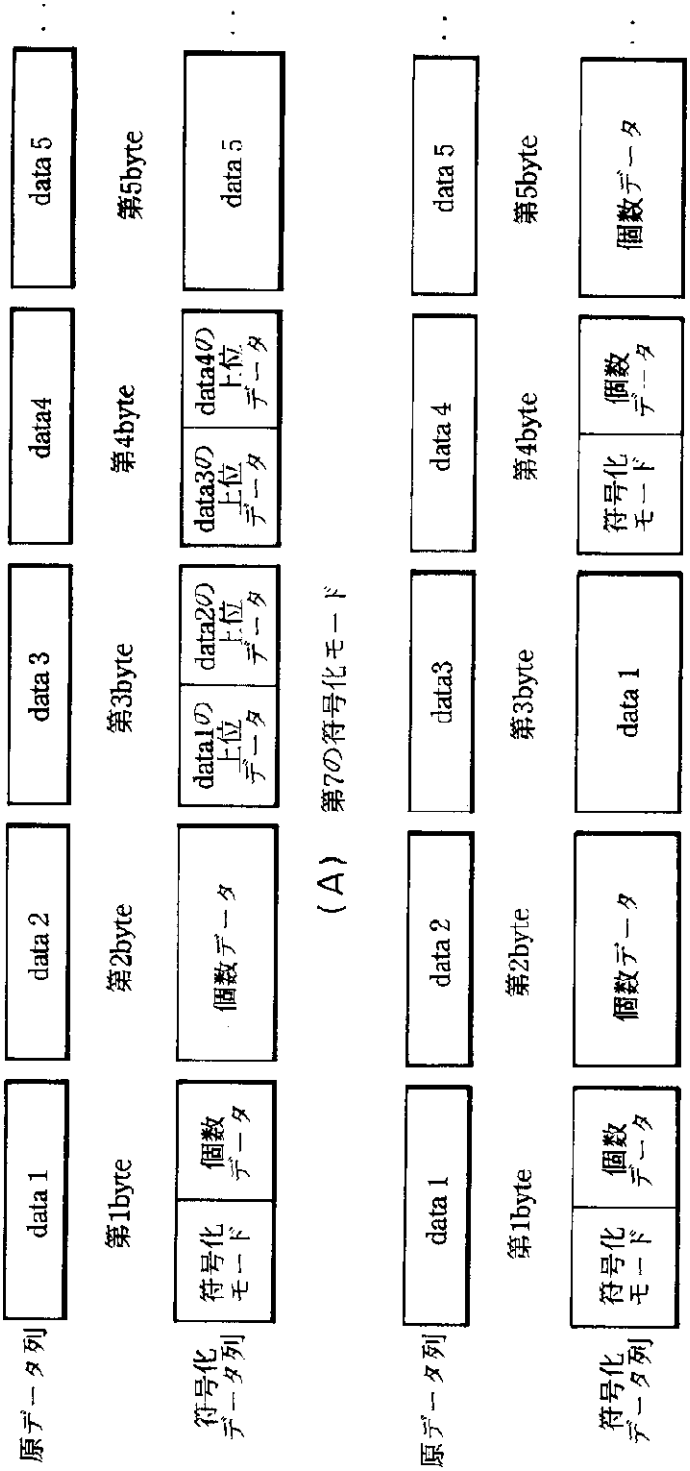
【図８】


復号化処理
第1byteの符号化データの下位データクリア
S401
第1byteの符号化データの上位データを、第1byteの復号化データの上位データとして保存
S402
符号化データの第2byteをNの最大値として保存
S403
カウントアップ
S404
S405
カウンタN>最大値？
YES
NO
S406
リターン
符号化データ中の(N+4)番目の4bitデータ読む
差分データ、下位データを分離
S407
復号化データの第Nbyteの上位データに対して差分データを加減し、第(N+1)byte目の上位データ復号化
S408
下位データとの第Nbyte目の上位データとの論理和を演算し、保存
S409

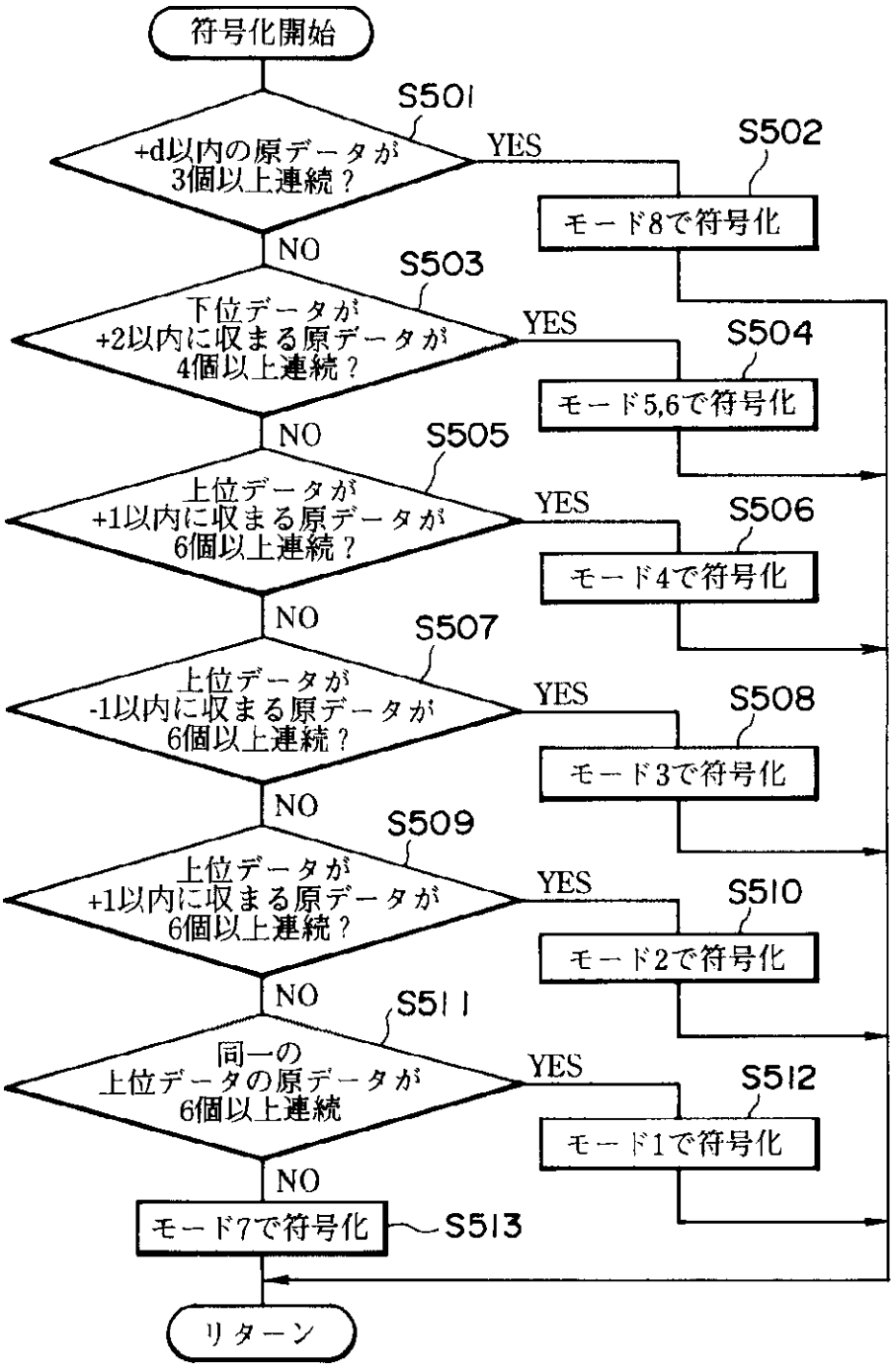
【図１１】


符号化データ列
符号化モード
個数データ
第1byte
個数データ
第2byte
符号化データ
第3byte
符号化データ
第4byte
. . .
(A) 第1～第4の符号モード
原データ列
data 1
第1byte
data 2
第2byte
data 3
第3byte
data4
第4byte
data 5
第5byte
. .
符号化データ列
符号化モード
個数データ
1
2
3
4
5
110
(B) 第5、第6の符号化モード

【図１２】


原データ列
data 1
data 2
data 3
data4
data 5
第1byte
第2byte
第3byte
第4byte
第5byte
符号化データ列
符号化モード
個数データ
個数データ
data1の上位データ
data2の上位データ
data3の上位データ
data4の上位データ
data 5
（A）　第7の符号化モード
原データ列
data 1
data 2
data3
data 4
data 5
第1byte
第2byte
第3byte
第4byte
第5byte
符号化データ列
符号化モード
個数データ
個数データ
data 1
符号化モード
個数データ
個数データ
（B）　第8の符号化モード

【図１３】


符号化開始
S501
+d以内の原データが
3個以上連続？
YES
S502
モード8で符号化
NO
S503
下位データが
+2以内に収まる原データが
4個以上連続？
YES
S504
モード5,6で符号化
NO
S505
上位データが
+1以内に収まる原データが
6個以上連続？
YES
S506
モード4で符号化
NO
S507
上位データが
-1以内に収まる原データが
6個以上連続？
YES
S508
モード3で符号化
NO
S509
上位データが
+1以内に収まる原データが
6個以上連続？
YES
S510
モード2で符号化
NO
S511
同一の
上位データの原データが
6個以上連続
YES
S512
モード1で符号化
NO
モード7で符号化
S513
リターン