

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5173045号
(P5173045)

(45) 発行日 平成25年3月27日 (2013. 3. 27)

(24) 登録日 平成25年1月11日 (2013.1.11)

(51) Int. Cl.		F I	
H03M 7/40	(2006.01)	H03M	7/40
H04N 1/41	(2006.01)	H04N	1/41 B
G10L 19/00	(2013.01)	G10L	19/00 400Z

請求項の数 10 (全 34 頁)

(21) 出願番号	特願2012-93208 (P2012-93208)	(73) 特許権者	000004226
(22) 出願日	平成24年4月16日 (2012. 4. 16)		日本電信電話株式会社
(62) 分割の表示	特願2011-514438 (P2011-514438) の分割		東京都千代田区大手町二丁目3番1号
原出願日	平成22年5月19日 (2010. 5. 19)	(74) 代理人	100121706
(65) 公開番号	特開2012-157067 (P2012-157067A)		弁理士 中尾 直樹
(43) 公開日	平成24年8月16日 (2012. 8. 16)	(74) 代理人	100128705
審査請求日	平成24年4月16日 (2012. 4. 16)		弁理士 中村 幸雄
(31) 優先権主張番号	特願2009-122088 (P2009-122088)	(74) 代理人	100147773
(32) 優先日	平成21年5月20日 (2009. 5. 20)		弁理士 義村 宗洋
(33) 優先権主張国	日本国 (JP)	(72) 発明者	原田 登
			東京都千代田区大手町二丁目3番1号 日 本電信電話株式会社内
		(72) 発明者	鎌本 優
			東京都千代田区大手町二丁目3番1号 日 本電信電話株式会社内

最終頁に続く

(54) 【発明の名称】 符号化方法、符号化装置、復号方法、復号装置、プログラム及び記録媒体

(57) 【特許請求の範囲】

【請求項1】

整数値系列の符号化方法であって、

[1] 0 以上かつ 0 以上の予め定めた値未満の整数（以下、規定整数という。）に対応する符号、並びに、[2] 上記予め定めた値以上の複数の整数の組に対応する符号（以下、拡張符号という。）、上記拡張符号に対応する符号化方法（以下、拡張符号化方法という。）が予め決められており、

上記拡張符号化方法は、整数値に割り当てられた符号の符号長が整数値の単調非減少関数値である符号化方法であり、

上記整数値系列中の符号化対象となる整数値が規定整数に該当する場合には、当該規定整数に対応する符号を符号化結果とし、

上記整数値系列中の符号化対象となる整数値が上記予め定めた値以上である場合には、上記予め定めた値が上記拡張符号化方法の入力の定義域の最小値に対応するように写像する写像関数で、上記符号化対象となる整数値を変換した値を、上記拡張符号化方法に入力して符号を得て、上記拡張符号と上記拡張符号化方法で得られた符号とを連結した符号を符号化結果とする、

符号化を行う符号化ステップ

を含む符号化方法。

【請求項2】

請求項1に記載の整数値系列の符号化方法であって、

入力信号の各サンプルが正負を含む整数で表されており、
 上記符号化ステップでは、上記入力信号中の各サンプルを0以上の整数に写像したものを上記符号化対象となる整数値として上記符号化ステップを行うことを特徴とする符号化方法。

【請求項3】

符号系列の復号方法であって、

[1]0以上かつ0以上の予め定めた値未満の整数(以下、規定整数という。)に対応する符号、並びに、[2]上記予め定めた値以上の複数の整数の組に対応する符号(以下、拡張符号という。)、上記拡張符号に対応する復号方法(以下、拡張復号方法という。)が予め決められており、

上記拡張復号方法は、整数値に割り当てられた符号の符号長が整数値の単調非減少関数値である復号方法であり、

上記符号系列中の符号が規定整数に対応する符号である場合には、当該符号に対応する規定整数を復号結果とし、

上記符号系列中の符号が上記予め定めた値以上の値に対応する符号である場合には、上記拡張復号方法を上記拡張符号に続く符号に対して適用して整数値を得て、上記拡張復号方法の出力の定義域の最小値が上記予め定めた値に対応するように写像する写像関数で、上記得られた整数値を変換した値を復号結果とする、

復号を行う復号ステップ

を含む復号方法。

【請求項4】

請求項3に記載の符号系列の復号方法であって、

上記復号結果である0以上の整数値を、正負を含む整数に写像したものを各サンプルとする出力信号を得る

ことを特徴とする復号方法。

【請求項5】

整数値系列の符号化装置であって、

[1]0以上かつ0以上の予め定めた値未満の整数(以下、規定整数という。)に対応する符号、並びに、[2]上記予め定めた値以上の複数の整数の組に対応する符号(以下、拡張符号という。)、上記拡張符号に対応する符号化方法(以下、拡張符号化方法という。)が予め決められており、

上記拡張符号化方法は、整数値に割り当てられた符号の符号長が整数値の単調非減少関数値である符号化方法であり、

上記整数値系列中の符号化対象となる整数値が規定整数に該当する場合には、当該規定整数に対応する符号を符号化結果とし、

上記整数値系列中の符号化対象となる整数値が上記予め定めた値以上である場合には、上記予め定めた値が上記拡張符号化方法の入力の定義域の最小値に対応するように写像する写像関数で、上記符号化対象となる整数値を変換した値を、上記拡張符号化方法に入力して符号を得て、上記拡張符号と上記拡張符号化方法で得られた符号とを連結した符号を符号化結果とする、符号化を行う符号化部

を含む符号化装置。

【請求項6】

請求項5に記載の整数値系列の符号化装置であって、

入力信号の各サンプルが正負を含む整数で表されており、

上記符号化部は、上記入力信号中の各サンプルを0以上の整数に写像したものを上記符号化対象となる整数値として上記符号化を行う

ことを特徴とする符号化装置。

【請求項7】

符号系列の復号装置であって、

[1]0以上かつ0以上の予め定めた値未満の整数(以下、規定整数という。)に対

10

20

30

40

50

応する符号、並びに、[2]上記予め定めた値以上の複数の整数の組に対応する符号（以下、拡張符号という。）、上記拡張符号に対応する復号方法（以下、拡張復号方法という。）が予め決められており、

上記拡張復号方法は、整数値に割り当てられた符号の符号長が整数値の単調非減少関数値である復号方法であり、

上記符号系列中の符号が規定整数に対応する符号である場合には、当該符号に対応する規定整数を復号結果とし、

上記符号系列中の符号が上記予め定めた値以上の値に対応する符号である場合には、上記拡張復号方法を上記拡張符号に続く符号に対して適用して整数値を得て、上記拡張復号方法の出力の定義域の最小値が上記予め定めた値に対応するように写像する写像関数で、上記得られた整数値を変換した値を復号結果とする復号部を含む復号装置。

10

【請求項 8】

請求項 7 に記載の符号系列の復号装置であって、

上記復号結果である 0 以上の整数値を、正負を含む整数に写像したものを各サンプルとする出力信号を得ることを特徴とする復号装置。

【請求項 9】

請求項 1 または請求項 2 に記載された符号化方法または請求項 3 または請求項 4 に記載された復号方法の各ステップをコンピュータに実行させるためのプログラム。

20

【請求項 10】

請求項 1 または請求項 2 に記載された符号化方法の各ステップをコンピュータに実行させるためのプログラムと、請求項 3 または請求項 4 に記載された復号方法の各ステップをコンピュータに実行させるためのプログラムとの少なくともいずれかのプログラムを記録した、コンピュータが読み取り可能な記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、信号の符号化技術に関し、より詳しくは、整数値系列の可逆符号化技術に関する。

30

【背景技術】

【0002】

音声周波数信号 (voice-frequency signals) に対する符号化 (encoding) の国際標準として、ITU-T (International Telecommunication Union - Telecommunication Standardization Sector) G.711 が存在する (非特許文献 1 参照)。ITU-T G.711 は、符号化方式として非線形 PCM (Non-uniform Pulse Code Modulation) を採用しており、非線形量子化方式として μ 則 (μ -law) と A 則 (A-law) の二種類を規定している。以下、「音声周波数信号」を音響信号と呼称する。

【0003】

また、動画 (Visual) と音響 (Audio) の符号化の国際標準として ISO/IEC JTC 1/SC 29/WG 11 が策定した MPEG-4 が存在し、その第 3 部が音響ロスレス符号化技術などを規定している (非特許文献 2 参照)。

40

【0004】

整数に対する符号化方法として、アルファ符号化 (unary coding) が知られている (非特許文献 3 参照)。アルファ符号化は、0 以上の整数 k を、 k 個の連続する整数 A とこれに続く 1 個の整数 B で表現する符号化方法である。整数 B は整数 A と異なり、整数 A が 1 の場合に整数 B は 0 とされ、整数 A が 0 の場合に整数 B は 1 とされる。

【0005】

例えば、 $k = 0$ のアルファ符号化による符号は 0 (または 1)、 $k = 1$ のアルファ符号化による符号は 1 0 (または 0 1)、 $k = 2$ のアルファ符号化による符号は 1 1 0 (また

50

は 0 0 1)、 $k = 3$ のアルファ符号化による符号は 1 1 1 0 (または 0 0 0 1)、 $k = 4$ のアルファ符号化による符号は 1 1 1 1 0 (または 0 0 0 0 1)、 $k = 5$ のアルファ符号化による符号は 1 1 1 1 1 0 (または 0 0 0 0 0 1) となる。

【 0 0 0 6 】

アルファ符号化の特徴として、(1) どんなに大きな整数 k であっても符号化可能である、(2) 符号化・復号の際に、符号化対象である整数 k と符号とを対応付けたテーブルを使わないので、テーブルを保存するためのメモリを必要としない、(3) 整数 k は、 $k + 1$ ビットで符号化される、(4) 大きな整数 k を表現する場合、整数 k を 2 進数で表現する場合と比較して多くのビットを必要とする、ということが挙げられる。

【 0 0 0 7 】

エントロピー符号化方法として、ハフマン符号化 (Huffman coding) が知られている (非特許文献 3 参照)。ハフマン符号化は、符号化対象である入力値の出現頻度に応じて可変長符号を割り当てる符号化方法であり、出現頻度がより高い入力値に対してより短い符号を与える。

【 0 0 0 8 】

ハフマン符号化の特徴として、(1) 符号化・復号の際に、符号化対象である入力値と符号とを対応付けたテーブルを使用する、(2) テーブルサイズによって符号化できる最大値が決まる (従って、符号化対象である入力値の種類が多ければ大きなテーブルが必要となる)、(3) テーブルサイズが大きい場合に、出現頻度の低い入力値のハフマン符号が、入力値を 2 進数で表現する符号化と比較して多くのビットを必要とする場合がある、

【 0 0 0 9 】

整数に対する符号化方法として、ゴロム ライス符号化 (Golomb-Rice coding) が知られている (非特許文献 3 参照)。0 以上の整数 t のゴロム ライス符号は、所定の整数 s (Rice パラメータ) を用いて式 (1) で求まる商 q をアルファ符号化し、式 (2) で求まる余り r を s ビットの二進符号で符号化して、さらに商 q のアルファ符号と余り r の二進符号を連結することで得られる。記号 \cdot は床関数を表す。

【 数 1 】

$$q = \left\lfloor \frac{t}{2^s} \right\rfloor \quad (1)$$

$$r = t - q \cdot 2^s \quad (2)$$

【 0 0 1 0 】

ゴロム ライス符号の特徴として、(1) ゴロム ライス符号の符号長は整数 s に依存する、(2) 整数 t のゴロム ライス符号は $q + s + 1$ ビットで表せる、(3) 商 q をアルファ符号化するため、整数 t が大きくなると整数 t のゴロム ライス符号の符号長は長くなる、(4) 入力値が正負の値を取る場合は、入力値を 0 以上の整数 t にしてからゴロム ライス符号化を行う、が挙げられる。

【 先行技術文献 】

【 非特許文献 】

【 0 0 1 1 】

【 非特許文献 1 】 ITU-T Recommendation G.711, "Pulse Code Modulation (PCM) of voice frequencies," ITU-T, 1993.

【 非特許文献 2 】 ISO/IEC 14496-3 AMENDMENT 2: Audio Lossless Coding (ALS), new audio profiles and BSAC extensions, 2005.

【 非特許文献 3 】 David Salomon, "Data Compression : The Complete Reference," 3rd edition, Springer-Verlag, ISBN-10: 0-387-40697-2, 2004.

【 発明の概要 】

【発明が解決しようとする課題】

【0012】

この明細書に用いる用語は、特に断りの無い限り、ITU-T G.701（参考文献1）に準拠する。しかし、これは本発明の適用範囲をITU-TのGシリーズ勧告が適用される技術分野に限定する意図ではない。この技術分野以外の技術分野、例えば上記非特許文献2に関わる技術分野に本発明を適用する場合には、この明細書に用いる用語を、（もし在れば）当該技術分野の対応する用語に読み替えればよい。

（参考文献1）ITU-T Recommendation G.701, "Vocabulary of Digital Transmission and Multiplexing, and Pulse Code Modulation (PCM) Terms," ITU-T, 1993.

【0013】

所定の複数のサンプルをまとめたフレーム単位で符号化処理を行う場合に、フレームに含まれるサンプル x_i （ $i = 1, 2, \dots, N$ ）の値の出現頻度分布がLaplace分布に従うならば、ゴロム ライス符号は、整数符号の中で最適符号である。これは、サンプル x_i （ $i = 1, 2, \dots, N$ ）の値の出現頻度分布がLaplace分布に従うならば、商 q の確率分布が（離散的に） $1/2, 1/4, 1/8, 1/16, \dots, 1/2^h, \dots$ となりアルファ符号が最適になることによる。

【0014】

フレームに含まれるサンプル x_i （ $i = 1, 2, \dots, N$ ）の値の出現頻度分布がLaplace分布に従わない場合、もしくは、統計的にサンプル数が十分ではない場合、ゴロム ライス符号は必ずしも最適符号にならない。これは、商 q の確率分布がアルファ符号で最適となる確率分布に従わないため、アルファ符号の符号長が大きくなることによる。

【0015】

ハフマン符号化は、符号化対象となるサンプル値の統計的な出現頻度に応じて、統計的に学習されたハフマン符号テーブルを用いることで、Laplace分布に限らず任意の出現頻度分布のサンプル値を高効率に符号化することができることが知られている。しかし、符号化対象となる整数値の取りうる範囲が広範にわたる場合には、大きなハフマン符号テーブルを具備する必要がある、ハフマン符号テーブルの記録容量が実施の制約になるという問題がある。

【0016】

そこで、本発明は、フレームに含まれるサンプル x_i （ $i = 1, 2, \dots, N$ ）の値の出現頻度分布がLaplace分布に従わない場合や統計的にサンプル数が十分ではない場合であっても、フレームの符号長の増加を抑えることのできる符号化とその復号に関わる技術を提供することを目的とする。

【課題を解決するための手段】

【0017】

本発明の符号化方法は、[1]0以上かつ0以上の予め定めた値未満の整数（以下、規定整数という。）に対応する符号、並びに、[2]前記予め定めた値以上の複数の整数の組に対応する符号（以下、拡張符号という。）、拡張符号に対応する符号化方法（以下、拡張符号化方法という。）が予め決められており、拡張符号化方法は、整数値に割り当てられた符号の符号長が整数値の単調非減少関数値である符号化方法であり、整数値系列中の符号化対象となる整数値が規定整数に該当する場合には、当該規定整数に対応する符号を符号化結果とし、整数値系列中の符号化対象となる整数値が前記予め定めた値以上である場合には、前記予め定めた値が拡張符号化方法の入力の定義域の最小値に対応するように写像する写像関数で、符号化対象となる整数値を変換した値を、拡張符号化方法に入力して符号を得て、拡張符号と拡張符号化方法で得られた符号とを連結した符号を符号化結果とする、符号化を行う符号化ステップを含む。

【0018】

本発明の復号方法は、[1]0以上かつ0以上の予め定めた値未満の整数（以下、規定整数という。）に対応する符号、並びに、[2]前記予め定めた値以上の複数の整数の組に対応する符号（以下、拡張符号という。）、拡張符号に対応する復号方法（以下、拡張復号

10

20

30

40

50

方法という。)が予め決められており、拡張復号方法は、整数値に割り当てられた符号の符号長が整数値の単調非減少関数値である復号方法であり、符号系列中の符号が規定整数に対応する符号である場合には、当該符号に対応する規定整数を復号結果とし、符号系列中の符号が前記予め定めた値以上の値に対応する符号である場合には、拡張復号方法を拡張符号に続く符号に対して適用して整数値を得て、拡張復号方法の出力の定義域の最小値が前記予め定めた値に対応するように写像する写像関数で、得られた整数値を変換した値を復号結果とする、復号を行う復号ステップを含む。

【0019】

上記符号化方法においては、入力信号の各サンプルが正負を含む整数で表されており、符号化ステップで、入力信号中の各サンプルを0以上の整数に写像したものを符号化対象となる整数値として上記符号化ステップを行い、上記復号方法においては、復号結果である0以上の整数値を、正負を含む整数に写像したものを各サンプルとする出力信号を得るようにしてもよい。

10

【発明の効果】

【0020】

本発明によれば、限定されたサイズの符号テーブルで広い範囲の入力を扱って高能率に符号化することができる。

また、本発明によれば、所定数のサンプルごとにフレームとして、複数の異なる出現頻度分布に対応した複数の符号テーブルの中から最適なものをフレームごとに選択的に用いることで、入力信号を高能率に符号化することができる。

20

また、本発明によれば、フレームの各サンプルの商がアルファ符号化されないので、フレームに含まれるサンプル x_i ($i = 1, 2, \dots, N$) の出現頻度分布がLaplace分布に従わない場合や統計的にサンプル数が十分ではない場合であっても、各サンプルの符号の符号長ひいてはフレームの符号長の増加を抑えることができる。

【図面の簡単な説明】

【0021】

【図1】実施形態1に係る符号化装置の機能構成例を示すブロック図。

【図2】実施形態1に係る符号化処理の処理フローを示す図。

【図3】実施形態1に係る復号装置の機能構成例を示すブロック図。

【図4】実施形態1に係る復号処理の処理フローを示す図。

30

【図5】実施形態2に係る符号化装置の機能構成例を示すブロック図。

【図6】実施形態2に係る符号化処理の処理フローを示す図。

【図7】実施形態2に係る復号装置の機能構成例を示すブロック図。

【図8】実施形態2に係る復号処理の処理フローを示す図。

【図9】実施形態4に係る符号化装置の機能構成例を示すブロック図。

【図10】商信号の値の分布例を示す図。

【図11】符号テーブルの例を示す図。

【図12】分離パラメータ $B = 0$ の場合に出力される誤差符号の例を示す図。

【図13】分離パラメータ $B > 0$ の場合に出力される誤差符号の例を示す図。

【図14】実施形態4に係る復号装置の機能構成例を示すブロック図。

40

【図15】図11に示した $n = 80$ の符号テーブル0に対応する復号テーブルの例を示す図。

【図16】実施形態4に係る復号処理の処理フローを示す図。

【図17】実施形態3に係る符号化装置の機能構成例を示すブロック図。

【図18】実施形態3に係る符号化処理の処理フローを示す図。

【図19】実施形態3に係る復号装置の機能構成例を示すブロック図。

【図20】実施形態3に係る復号処理の処理フローを示す図。

【図21】符号テーブルの一例(表1)。

【図22】符号テーブルの一例(表2)。

【図23】符号テーブルの一例(表3)。

50

【図 2 4】符号テーブルの一例（表 4）。

【図 2 5】符号テーブルの一例（表 5）。

【図 2 6】符号テーブルの一例（表 6）。

【図 2 7】符号テーブルの一例（表 7）。

【図 2 8】符号テーブルの一例（表 8）。

【発明を実施するための形態】

【0022】

< 本発明の実施形態 1 の符号化の概要 >

実施形態 1 で扱う入力信号は 0 以上の整数値系列とする。この整数値系列は、例えば、上記非特許文献 1 の音響口スレス符号化技術で行っているように、音響信号に線形予測分析を適用して得られる線形予測残差の値を、振幅の絶対値が小さなものが小さな値に割り当てられるように 0 以上の整数値へ一意の線形写像を行って得られる整数値系列であっても良い。

10

【0023】

本発明の実施形態 1 の符号化は、入力信号が 0 以上の整数値系列であり、且つ、入力信号の整数値が、特定の範囲の値において比較的出現確率が高いという統計的な性質を有する場合に、高能率に圧縮することができる。

【0024】

符号化装置は、[1]整数（以下、規定整数という。）とこれに対応する符号との組み合わせ、並びに、[2]規定整数以外の複数の整数による組とこの組に対応する一つの符号（以下、拡張符号という。）との組み合わせを少なくとも 1 つ、を記録した符号テーブルを具備し、上記入力信号の整数値系列中の符号化対象となる整数値を逐次符号化して符号を出力する。規定整数以外の整数を以下では規定外整数という。

20

【0025】

この符号化処理において、上記入力信号の整数値系列中の符号化対象となる整数値が、規定整数に該当する場合には、当該規定整数に対応する符号を上記符号テーブルから選択して符号化結果の符号とし、上記整数値系列中の符号化対象となる整数値が、規定外整数に該当する場合には、上記符号テーブルに記録された上記拡張符号と、あらかじめ定められていた上記拡張符号に対応する符号化方法を上記規定外整数に対して適用して得られる符号とを連結した符号を符号化結果の符号とする符号化が行われる。

30

【0026】

実施形態 1 の符号化装置 600E の機能的構成例を図 1 に示す。

符号化装置 600E は、符号テーブル 622 を記憶する記憶部 620E と、整数値系列を入力として符号テーブル 622 を用いて符号系列を出力する符号化部 610E とを含む。

符号化処理手順を図 2 に示す。

まず、未処理の整数値が存在するか否か判定する（ステップ S600）。存在しないならば処理を終了する。存在すれば、整数値系列のうち、未処理の整数値の先頭のを X とする（ステップ S602）。そして、符号テーブル 622 に X が規定整数として登録されているか否かを検索する（ステップ S604）。X が規定整数か否か判定する（ステップ S606）。X が規定整数ならば、符号テーブル 622 に登録されている規定整数 X に対応する符号を出力する（ステップ S608）。X が規定整数でないならば、拡張符号と拡張符号に対応する符号化方式で X を符号化した符号を結合して出力する（拡張符号が複数存在する場合には、最も符号量が少なくなるものを用いる）（ステップ S610）。

40

【0027】

符号テーブル 622 の一例を表 1 ~ 表 6 に示す。

表 1 ~ 表 6 に示す各符号テーブルでは、記号：の右側の符号が、その左側の規定整数に対応付けられている。規定外整数 p に対応する符号は、拡張符号と、拡張符号で特定される符号化方法で規定外整数 p を符号化した結果の符号と、の連結で得られる。記号 は、符号同士の連結を表す。

50

【 0 0 2 8 】

拡張符号によって特定される、拡張符号に対応する符号化方式は、符号化装置と復号化装置のそれぞれにおいてあらかじめ決定しておく。なお、可逆符号化を行うためには、符号化装置と復号化装置とで同一の符号化方式を用いる必要があるため、拡張符号に対応する符号化方式は、符号化装置と復号化装置との間で一意に決定しておく。以下、表 1 ~ 表 6 の例では、簡便のために、U はアルファ符号化を、R はゴロム ライス符号化を、F は固定長符号化を表す記号として用い、それらの符号化方法で数値 k を符号化して得られた符号を $\text{code}(U, k)$ 、 $\text{code}(R, s, k)$ 、 $\text{code}(F, g, k)$ 等であらわすこととする。

【 0 0 2 9 】

表 1 の例 (図 2 1 参照) では、1 種類の拡張符号が規定されており 1 1 1 1 である。拡張符号 1 1 1 1 に対応した符号化方式はアルファ符号 $\text{code}(U, p-4)$ であり、あらかじめ符号化装置と復号化装置の間で一意に決定しておく。

$\text{code}(U, p-4)$ は、符号化対象となる整数値が規定外整数 p であるとき、整数 p から 4 を引いた $p - 4$ をアルファ符号化して得られる符号をあらわしている。但し、この例では $p = 4$ である。

入力信号中の符号化対象となる整数値が規定整数に相当する 0 ~ 3 あれば、表 1 に登録された 0 ~ 3 に対応する符号が選択されて出力される。たとえば、規定整数である整数値 3 の符号は 1 1 1 0 となる。

また、規定外整数 $p = 6$ の符号は拡張符号 1 1 1 1 と $p - 4$ をアルファ符号化して得られる 1 1 1 0 を連結した 1 1 1 1 1 1 0 となる。

【 0 0 3 0 】

表 2 の例 (図 2 2 参照) では、2 種類の拡張符号が規定されており 1 1 1 1 0 および 1 1 1 1 1 である。拡張符号 1 1 1 1 0 に対応した符号化方式はアルファ符号で、拡張符号 1 1 1 1 1 に対応した符号化方式は固定長符号とライス符号との連結である。拡張符号と対応する符号化方式の組み合わせは、あらかじめ符号化装置と復号化装置の間で一意に決定しておく。

この例の $\text{code}(U, p-4)$ は、拡張符号 1 1 1 1 0 に対応する符号化方式が出力する符号で、符号化対象となる整数値が規定外整数 p であるとき、整数 p から 4 を引いた $p - 4$ をアルファ符号化して得られる符号をあらわしている。 $\text{code}(F, g, s-1)$ $\text{code}(R, s, p-4)$ は、拡張符号 1 1 1 1 0 に対応する符号化方式が出力する符号であり、符号化対象となる整数値が規定外整数 p であるとき、Rice パラメータ s の g ビットの固定長符号 $\text{code}(F, g, s-1)$ と、Rice パラメータ s のゴロム・ライス符号化によって得られる整数 $p - 4$ の符号 $\text{code}(R, s, p-4)$ を連結した符号を表す。但し、 $p = 4$ である。採りえる s の値の個数 W について $g = \log_2 W$ である。また、 $\text{code}(R, s, p-4)$ は、 $p-4$ の二進表現の下位 s ビットを出力した後に、 $(p-4)/(2^s)$ の商をアルファ符号で出力するものとする。この順序は符号化処理部と、復号処理部の間であらかじめ一意に決めておいたものを用いる。

例えば Rice パラメータ s が 1 $s = 4$ と設定されている場合、 $g = 2$ であり、規定外整数 $p = 12$ の符号は

1 1 1 1 0 1 1 1 1 1 1 1 1 0 または

1 1 1 1 1 0 0 0 1 1 1 1 0 ($s = 1$; 2 ビット符号 0 0) または

1 1 1 1 1 0 1 0 0 1 1 0 ($s = 2$; 2 ビット符号 0 1) または

1 1 1 1 1 1 0 0 0 1 0 ($s = 3$; 2 ビット符号 1 0) または

1 1 1 1 1 1 1 1 0 0 0 0 ($s = 4$; 2 ビット符号 1 1) となる。

2 種類以上の拡張符号が規定されている場合には、少なくとも 2 つの符号が得られるから、規定外整数の符号は、得られた符号のうち最小の符号長を持つものとする (上記の例では、 $s = 2, 3, 4$ のいずれでも符号長は同じになるので、いずれでもよい。たとえば最初のエントリーの $s = 2$ を用いる。)

【 0 0 3 1 】

表 3 の例 (図 2 3 参照) では、4 種類の拡張符号が規定されており 1 1 1 1 0 0、1 1 1 1 0 1、1 1 1 1 1 0、1 1 1 1 1 1 である。 $\text{code}(U, p-4)$ は、アルファ符号化によっ

10

20

30

40

50

て得られる整数 $p - 4$ の符号を表す。code(F,g,s-3)は、Riceパラメータ s の g ビットの固定長符号を表す。採りえる s の値の個数 W について $g = \log_2 W$ である。code(R,s,p-4)はRiceパラメータ s のゴロム・ライス符号化によって得られる整数 $p - 4$ の符号を表す。但し、 $p - 4$ である。拡張符号が 1 1 1 1 0 1 の場合はRiceパラメータ s が 1、拡張符号が 1 1 1 1 1 0 の場合はRiceパラメータ s が 2 に固定されている。

この例では、拡張符号が 1 1 1 1 1 1 の場合には、Riceパラメータ s は 3、 $s = 6$ の 4 通りの値をとり、 $s - 3$ を $g = 2$ ビットの固定長符号 code(F,g,s-3) であらわし、固定長符号 code(F,g,s-3) と Riceパラメータ s を用いたゴロム・ライス符号により $p - 4$ を符号化した符号 code(R,s,p-4) と連結する。

たとえば $p = 12$ の符号は

1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 または
 1 1 1 1 0 1 0 1 1 1 1 0 ($s = 1$) または
 1 1 1 1 1 0 0 0 1 1 0 ($s = 2$) または
 1 1 1 1 1 1 0 0 0 0 1 0 ($s = 3$; 2 ビット符号 0 0) または
 1 1 1 1 1 1 0 1 1 0 0 0 0 ($s = 4$; 2 ビット符号 0 1) または
 1 1 1 1 1 1 1 0 0 0 1 0 0 0 ($s = 5$; 2 ビット符号 1 0) または
 1 1 1 1 1 1 1 1 0 0 0 1 0 0 0 ($s = 6$; 2 ビット符号 1 1) となる。

2 種類以上の拡張符号が規定されている場合には、少なくとも 2 つの符号が得られるから、規定外整数の符号は、得られた符号のうち最小の符号長を持つものとする。

【 0 0 3 2 】

表 4 の例 (図 2 4 参照) では、2 種類の拡張符号が規定されており 1 1 1 1 0、1 1 1 1 1 である。code(U,p-4:4 p 6)はアルファ符号化によって得られる整数 $p - 4$ (但し、 $4 \leq p \leq 6$) の符号を表す。code(U,p-12:15 p)はアルファ符号化によって得られる整数 $p - 12$ (但し、 $15 \leq p$) の符号を表す。code(F,h,p-7:7 p 14)は、固定長符号化によって得られる整数 $p - 7$ の h ビット符号を表す。採りえる p の値の個数 Q について $h = \log_2 Q$ である。 p が $7 \leq p \leq 14$ を満たす整数である場合、 $h = 3$ である。表 4 の例は、小さい整数ほどその出現頻度 (出現確率) が高い場合に、これらの整数にできるだけ短い符号を割り当てるように工夫したものであり、表 1 の例の符号テーブルを改良したものに相当する。例えば $p (4 \leq p \leq 19)$ の符号は

1 1 1 1 0 0 ($p = 4$)、
 1 1 1 1 0 1 0 ($p = 5$)、
 1 1 1 1 0 1 1 0 ($p = 6$)、
 1 1 1 1 1 0 0 0 ($p = 7$)、
 1 1 1 1 1 0 0 1 ($p = 8$)、
 1 1 1 1 1 0 1 0 ($p = 9$)、
 1 1 1 1 1 0 1 1 ($p = 10$)、
 1 1 1 1 1 1 0 0 ($p = 11$)、
 1 1 1 1 1 1 0 1 ($p = 12$)、
 1 1 1 1 1 1 1 0 ($p = 13$)、
 1 1 1 1 1 1 1 1 ($p = 14$)、
 1 1 1 1 0 1 1 1 0 ($p = 15$)、
 1 1 1 1 0 1 1 1 1 0 ($p = 16$)、
 1 1 1 1 0 1 1 1 1 1 0 ($p = 17$)、
 1 1 1 1 0 1 1 1 1 1 1 0 ($p = 18$)、
 1 1 1 1 0 1 1 1 1 1 1 1 0 ($p = 19$) である。

【 0 0 3 3 】

表 5 の例 (図 2 5 参照) では、3 種類の拡張符号が規定されており 1 1 1 1 0 1、1 1 1 1 1 0、1 1 1 1 1 1 である。code(U,p-5)はアルファ符号化によって得られる整数 $p - 5$ の符号を表す。code(F,3,p-5)は固定長符号化によって得られる整数 $p - 5$ の 3 ビット符号を表す。code(F,4,p-5)は固定長符号化によって得られる整数 $p - 5$ の 4 ビット符

10

20

30

40

50

号を表す。但し、 $p = 5$ である。

【0034】

表6の例(図26参照)では、2種類の拡張符号が規定されており1110、110である。code(U,p-3:p=3)はアルファ符号化によって得られる整数 $p - 3$ の符号0を表す。code(U,p-4:5 p)はアルファ符号化によって得られる整数 $p - 4$ (但し、 $5 \leq p$)の符号を表す。code(F,3,p-4:5 p)は、固定長符号化によって得られる整数 $p - 4$ (但し、 $5 \leq p$)の3ビット符号を表す。表1~表5の例では、規定外整数 p の符号が規定整数の符号の符号長よりも長い符号長を持つものであったが、この表6の例のように、規定外整数 p の符号の符号長が規定整数の符号の符号長と同じか短くなるように規定外整数 p の符号を割り当てることもできる。規定整数の出現頻度と比較して、規定外整数の出現頻度が相対的に高い場合に、規定外整数に対応する拡張符号長が短い符号テーブルを用いる。このように、符号テーブルは、あらかじめ統計的な出現頻度の期待値を計測して設計する。

10

【0035】

<本発明の実施形態1の復号化の概要>

本発明の実施形態1の復号化では、実施形態1の符号化により生成された符号系列を入力とし、元の整数値系列を得る。

符号系列中の符号は元の整数値に対応した可変長符号化により符号化されており、先頭から逐次復号することによって元の整数系列を得る。

【0036】

実施形態1の復号装置650Dの機能的構成例を図3に示す。

20

復号装置650Dは、符号テーブル622を記憶する記憶部670Dと、符号系列を入力として符号テーブル622を用いて整数値系列を出力する復号部660Dを含む。

復号処理手順を図4に示す。

まず、未処理の符号が存在するか否か判定する(ステップS650)。存在しないならば処理を終了する。存在すれば、符号系列のうち、未処理の符号の先頭のビットを処理開始点とする(ステップS652)。そして、符号テーブル622に登録されている符号のうち、符号系列と一致するものを探索する(ステップS654)。規定整数に対応する符号であるか否かを判定する(ステップS656)。規定整数に対応する符号である場合、符号テーブル622に登録されている符号に対応する規定整数を整数値Xとして出力する(ステップS658)。規定整数に対応する符号でない場合、拡張符号と拡張符号に対応する符号化方式を判別し、判別された符号化方式で拡張符号に続く符号を復号して得られる整数値Xを出力する(ステップS660)。

30

【0037】

つまり、符号系列の先頭のビットから順に、符号テーブルに登録されている規定整数に対応する符号、もしくは、拡張符号と特定できるまで、符号ビットを逐次読み込む。

読み込んだ符号が、規定整数に対応する符号と一致した場合には、当該符号に対応する規定整数を復号された整数値として出力する。

読み込んだ符号が、拡張符号である場合には、拡張符号に対応してあらかじめ一意に定められた符号化方式の符号を、拡張符号に続く符号系列から読み込んで復号し、結果の整数値を出力する。

40

【0038】

例えば、表2(図22参照)を用いて符号化された符号列から、元の整数値系列を復号する場合に、入力符号系列が以下の場合を例にとって説明する。

010111101011111010110

ただし、 $g=2$ とする。

図4のステップに従えば、未処理の符号を先頭のビットより逐次調べて、表2に登録された符号中の一致する符号を見つける。表2より、上記入力符号系列は以下のように解析される。

0, 10, 11110 10, 11111 01 01 10

したがって、復号により得られる整数値系列は、0, 1, 5, 5, 9となる。

50

【 0 0 3 9 】

< 本発明の実施形態 2 の符号化の概要 >

本発明の実施形態 2 の符号化装置 7 0 0 E の機能的構成例を図 5 に示す。

符号化装置 7 0 0 E は、複数の符号テーブル 7 2 2 を記憶する記憶部 7 3 0 E と、整数値系列を入力として所定のサンプル数ごとにまとめたフレームのサンプルを出力するフレーム分割部 7 1 0 E と、フレームのサンプルを入力として符号テーブル 7 2 2 を用いて符号系列と符号テーブル指定符号を出力する符号化部 7 2 0 E を含む。

【 0 0 4 0 】

符号化処理を図 6 に示す。

まず、未処理のフレームが存在するか否か判定する（ステップ S 7 0 0 ）。存在しないならば処理を終了する。存在すれば、当該フレームに用いる符号テーブルを選択し符号テーブル指定符号を出力する（ステップ S 7 0 2 ）。それぞれの符号テーブルを用いて整数値系列を符号化した場合の符号量を推定し、最も符号量が少なくなる符号テーブルを選択する。また各符号テーブルを用いた場合の符号量を推定して符号テーブルを選択することとしてもよい。次に、選択された符号テーブルを用いて整数値系列の符号化処理を行う（ステップ S 7 0 4 ）。フレーム内に未処理の整数値が存在するか否かを判定する（ステップ S 7 0 6 ）。フレーム内に未処理の整数値が存在しない場合、ステップ S 7 0 0 の処理を行う。フレーム内に未処理の整数値が存在する場合、整数値系列のうち、未処理の整数値の先頭のを X とする（ステップ S 7 0 8 ）。そして、符号テーブル 7 2 2 に X が規定整数として登録されているか否かを検索する（ステップ S 7 1 0 ）。X が規定整数か否か判定する（ステップ S 7 1 2 ）。X が規定整数ならば、符号テーブル 7 2 2 に登録されている規定整数 X に対応する符号を出力する（ステップ S 7 1 4 ）。X が規定整数でないならば、拡張符号と拡張符号に対応する符号化方式で X を符号化した符号を結合して出力する（拡張符号が複数存在する場合には、最も符号量が少なくなるものを用いる）（ステップ S 7 1 6 ）。ステップ S 7 1 4 と S 7 1 6 の後、ステップ S 7 0 6 の処理を行う。

【 0 0 4 1 】

入力される整数値系列中の整数値を、先頭から逐次、あらかじめ定めておいた個数ごとにフレームとしてまとめて処理する点、および、複数の符号テーブルを備え、フレーム単位でフレームの符号量が最も小さくなるように符号テーブルを切り替える点、選択された符号テーブルを特定する符号テーブル指定符号をフレームごとに出力する点が実施形態 1 と異なる。

【 0 0 4 2 】

< 本発明の実施形態 2 の復号の概要 >

本発明の実施形態 2 の復号装置 7 5 0 D の機能的構成例を図 7 に示す。

復号装置 7 5 0 D は、複数の符号テーブル 7 2 2 を記憶する記憶部 7 8 0 D と、符号テーブル指定符号を入力として符号テーブル 7 2 2 の選択制御を行う符号テーブル選択制御部 7 6 0 D と、符号系列を入力として符号テーブル 7 2 2 を用いて整数値系列を出力する復号部 7 7 0 D を含む。

【 0 0 4 3 】

復号処理を図 8 に示す。

まず、未処理の符号が存在するか否か判定する（ステップ S 7 5 0 ）。存在しないならば処理を終了する。存在すれば、符号系列から符号テーブル指定符号を読み込み、符号テーブルを選択する（ステップ S 7 5 2 ）。そして、フレーム内の整数値の復号が完了したかを判定する（ステップ S 7 5 4 ）。完了した場合は、ステップ S 7 5 0 の処理を行う。完了していない場合、符号系列のうち、未処理の符号の先頭のビットを処理開始点とする（ステップ S 7 5 6 ）。そして、符号テーブル 7 2 2 に登録されている符号のうち、符号系列と一致するものを探索する（ステップ S 7 5 8 ）。規定整数に対応する符号であるかを判定する（ステップ S 7 6 0 ）。規定整数に対応する符号である場合、符号テーブル 7 2 2 に登録されている符号に対応する規定整数を整数値 X として出力する（ステップ S 7 6 2 ）。規定整数に対応する符号でない場合、拡張符号と拡張符号に対応する符号化

10

20

30

40

50

方式を判別し、判別された符号化方式で拡張符号に続く符号を復号して得られる整数値 X を出力する（ステップ S 7 6 4）。

【 0 0 4 4 】

実施形態 2 の復号処理は、フレーム単位で復号処理を行う。

まず、符号テーブル指定符号を復号し、指定された符号テーブル（もしくは符号テーブルに対応する復号テーブル）を用いて復号する点が実施形態 1 の復号方法と異なる。

符号系列の先頭のビットから順に、符号テーブルに記録されている規定整数に対応する符号、もしくは、拡張符号と特定できるまで、符号ビットを逐次読み込む。

読み込んだ符号が、規定整数に対応する符号と一致した場合には、当該符号に対応する規定整数を復号された整数値として出力する点は実施形態 1 の復号方法と同じである。

10

【 0 0 4 5 】

< 本発明の実施形態 3 の符号化の概要 >

上記の実施形態 1 および 2 を、ゴロム ライス符号化に類似の符号化と組み合わせた実施形態を以下に示す。

本実施形態で扱う入力信号は、音響信号に何らかの符号化方式を適用して符号化することで得られた信号とする。但し、入力信号の各サンプルは整数で表されているとする。このような符号化方式として、例えば上記非特許文献 1 の I T U - T G . 7 1 1 などが挙げられる。また、入力信号は、音響信号に何らかの符号化方式を適用して符号化することで得られた信号（元信号）そのものに限定されず、例えば、元信号に線形予測分析を適用して得られる線形予測残差であってもよい。

20

【 0 0 4 6 】

符号化装置は、入力信号をフレーム単位で符号化する。まず、符号化装置の分離パラメータ計算部が、1 フレームの入力信号から当該フレームの各サンプルに対する分離パラメータ B を求める。分離パラメータ B は様々な方法で決定することができる。たとえば本実施形態 3 ではゴロム ライス符号化の Rice パラメータと類似の方法で決定する。

【 0 0 4 7 】

Rice パラメータ s の算出について説明する。Rice パラメータ s は、1 フレームの入力信号の各サンプルをゴロム ライス符号化して得られる当該入力信号の符号長（つまり、各サンプルのゴロム ライス符号の合計符号長）を最小にするように定められる。なお、入力信号が例えば符号化方式 I T U - T G . 7 1 1 で得られている場合、この入力信号の各サンプルは正負の整数で表される場合がある。このように各サンプルが正負の整数で表される場合には、Rice パラメータ s は、0 以上の整数に変換した 1 フレームの入力信号の各サンプルを符号化して得られる当該入力信号の符号長（つまり、各サンプルのゴロム ライス符号の合計符号長）を最小にするように定められる。以下の例では、入力信号の各サンプルが正負の整数で表されているとする。

30

【 0 0 4 8 】

入力信号に含まれる或るサンプル x_i ($i = 1, 2, \dots, N$; N は 1 フレームに含まれるサンプル数) について、0 以上の整数に変換したこのサンプル x_i の Rice パラメータ s を用いた商 q_i は式 (3)、式 (4) で与えられる。式 (3) により商 q_i を求める場合は、余りを表す符号を、余りを表す $s - 1$ ビットの二進符号にサンプル x_i の正負を識別するための 1 ビットの符号が付加された s ビットの固定長を持つ符号とする。式 (4) により商 q_i を求める場合は、商 q_i を表す符号の L S B (Least Significant Bit) によってサンプル x_i の正負が識別される。

40

【数2】

 $s \geq 1$ の場合

$$q_i = \begin{cases} \left\lfloor \frac{x_i}{2^{s-1}} \right\rfloor = \left\lfloor \frac{2x_i}{2^s} \right\rfloor & (x_i \geq 0) \\ \left\lfloor \frac{-x_i-1}{2^{s-1}} \right\rfloor = \left\lfloor \frac{-2x_i-2}{2^s} \right\rfloor & (x_i < 0) \end{cases} \quad (3)$$

10

 $s = 0$ の場合

$$q_i = \begin{cases} \lfloor 2x_i \rfloor = \left\lfloor \frac{2x_i}{2^s} \right\rfloor & (x_i \geq 0) \\ \lfloor -2x_i - 1 \rfloor = \left\lfloor \frac{-2x_i - 1}{2^s} \right\rfloor & (x_i < 0) \end{cases} \quad (4)$$

【0049】

或るサンプル x_i を表す符号の符号長 C は式 (5) で得られる。ただし、 a は 0 または 1 または 2 である。記号 $\lfloor \cdot \rfloor$ は絶対値を表す。式 (5) の第 1 項と第 2 項の合計は商 q_i のアルファ符号の符号長を表し、第 3 項は余りを表す二進符号の符号長を表している。

20

【数3】

$$C = \left\lfloor \frac{2|x_i| - a}{2^s} \right\rfloor + 1 + s \quad (5)$$

【0050】

従って、0 以上の整数に変換された入力信号の符号長、つまり 0 以上の整数に変換された各サンプルのゴロム ライス符号の符号長の合計 $C(s, x_i, N)$ は式 (6) で与えられる。なお、式 (6) に現れる D は式 (7) で与えられる。

【数4】

$$C(s, x_i, N) = \sum_{i=1}^N \left\{ \left\lfloor \frac{2|x_i| - a}{2^s} \right\rfloor + 1 + s \right\} \\ = \frac{1}{2^s} (2D - aN) + (1 + s)N \quad (6)$$

30

$$D = \sum_{i=1}^N |x_i| \quad (7)$$

40

【0051】

$C(s, x_i, N)$ を最小にする s (以下、 s_0 とする。) は、式 (8) を s について解くことで得られる。つまり s_0 は、式 (9) で与えられる。記号 \ln は自然対数を表す。所望の Rice パラメータ s は、 $s = s_0$ または $s = s_0 + 1$ で得られる。記号 $\lceil \cdot \rceil$ は天井関数を表す。なお、数値計算を実行することによる処理速度の低下を避けるため、予め D 、 N 、 a との組み合わせで s_0 を求めておき、 D 、 N 、 a の組み合わせに s を対応付けたテーブルを記憶部に記憶しておいてもよい。 D 、 N 、 a が定まればテーブルを参照することで Rice パラメータ s を決定できるから処理速度の低下を回避できる。テーブルに D 、 N 、 a の組み合わせが無い場合には数値計算で s_0 を求めればよい。

【数5】

$$\frac{\partial C(s, x_i, N)}{\partial s} = -\frac{\ln 2}{2^s} (2D - aN) + N = 0 \quad (8)$$

$$s_0 = \log_2 \left(\ln 2 \cdot \left(\frac{2D}{N} - a \right) \right) \quad (9)$$

【0052】

D / N が a よりも十分に大きい場合、式(9)は式(10)に書き改められる。

【数6】

$$s_0 = \log_2 \left(\ln 2 \cdot \frac{2D}{N} \right) \quad (10)$$

10

【0053】

フレームのライス符号長 $C(s, x_i, N)$ の最小値 $C(x_i, N)$ は式(11)で与えられる。

【数7】

$$C(x_i, N) = N \left[\frac{1}{\ln 2} + 1 + \log_2 \left(\ln 2 \cdot \left(\frac{2D}{N} - a \right) \right) \right] \quad (11)$$

【0054】

結局、分離パラメータ計算部は、式(9)または式(10)を用いて1フレームの入力信号からRiceパラメータ $s = s_0$ または $s = s_0$ を求めることで、分離パラメータ B を求める。

20

【0055】

分離パラメータ B が求まると、符号化装置の商余計算部は、1フレームの入力信号の各サンプルを 0 以上の整数に変換してその商 y_i と余り z_i を求める。商 y_i の算出は、分離パラメータ B に応じて式(12)または式(13)に従う。B = 0 の場合、除数は $2^0 = 1$ であるため、商 y_i の算出は、実質的には、サンプル x_i が 0 以上であるならば偶数に置換し、サンプル x_i が負であるならば奇数に置換する処理に相当する。

【数8】

$B \geq 1$ の場合

30

$$y_i = \begin{cases} \left\lfloor \frac{x_i}{2^{B-1}} \right\rfloor & (x_i \geq 0) \\ \left\lfloor \frac{-x_i - 1}{2^{B-1}} \right\rfloor & (x_i < 0) \end{cases} \quad (12)$$

$B = 0$ の場合

$$y_i = \begin{cases} 2x_i & (x_i \geq 0) \\ -2x_i - 1 & (x_i < 0) \end{cases} \quad (13)$$

40

【0056】

余り z_i は、式(14)で得られる。B = 0 の場合、余り z_i は得られない。

【数 9】

$$z_i = \begin{cases} x_i - \left\lfloor \frac{x_i}{2^{B-1}} \right\rfloor \cdot 2^{B-1} & (x_i \geq 0) \\ -x_i - 1 - \left\lfloor \frac{-x_i - 1}{2^{B-1}} \right\rfloor \cdot 2^{B-1} & (x_i < 0) \end{cases} \quad (14)$$

【0057】

余り z_i は、B ビットの二進符号で表現される。商 y_i を式 (12) で得る場合、余り z_i は B - 1 ビットの二進符号で表されるが、サンプル x_i の正負を識別するための 1 ビットを付加して B ビットの二進符号としている。例えば、この識別用の 1 ビットを余り z_i を表す二進符号の MSB (Most Significant Bit) とし、サンプル x_i が 0 以上であれば 1 とし、サンプル x_i が 0 未満であれば 0 とする。

10

【0058】

符号化装置の符号化部は、少なくとも一つの符号テーブルを用いて商 y_i を符号化する。符号テーブルはそれぞれ、(1) 整数とこれに対応する符号との組み合わせ、並びに (2) 前記組み合わせに規定される整数以外の複数の整数の組とこの組に対応する少なくとも一つの符号 (以下、拡張符号という。) との組み合わせを記録したデータベースである。なお、ここで整数は 0 以上の値を持つ。以下、上記組み合わせに規定される整数を規定整数と云い、規定整数以外の整数を規定外整数と云い、拡張符号に対応する符号化方法を拡張符号化方法と云うことにする。

20

【0059】

商符号化部は、符号テーブルを用いて、商 y が規定整数のいずれかに相当する場合は、その対応する符号を出力し、商 y が規定外整数に相当する場合は、拡張符号と、拡張符号に対応する予め定められた符号化方法で得られる規定外整数 p の符号とを連結した符号を出力する。

【0060】

上記の各表 1 - 6 に示した符号テーブルの例において、規定整数および規定整数に対応する符号は例示に過ぎない。これらは、好ましくは、事前に用意された入力信号 (学習用入力信号) を用いた学習によって設定される。例えば、学習用入力信号から得られる Rice パラメータ s を用いて 0 以上の整数に変換された学習用入力信号の各サンプルの商を求め、これら商のハフマン符号を作成する。これら商のうち出現頻度 (出現確率) の高いものとそのハフマン符号が規定整数とその符号として採用される。拡張符号は、規定整数の符号とは異なる二進符号であればよく、任意に設定される。拡張符号に対応する符号化方法も任意に設定される。

30

【0061】

また、学習に用いる Rice パラメータ s を学習用入力信号から得られる Rice パラメータ s に限定せず、任意に定めた 2 種類以上のパラメータ s (好ましくは、学習用入力信号から得られる Rice パラメータ s に近い値とする。) を用いて、パラメータ s ごとに、0 以上の整数に変換された学習用入力信号の各サンプルの商とこれら商の符号 (例えばハフマン符号) を作成することで、1 種類の学習用入力信号から複数の符号テーブルを作成してもよい。

40

【0062】

学習用入力信号から得られる Rice パラメータ s と学習用入力信号の各サンプルの商の出現頻度 (出現確率) は、学習用入力信号の統計的性質に依存するから、学習用入力信号を複数用意し、各学習用入力信号から複数の符号テーブルを作成してもよい。

【0063】

符号化対象の入力信号の統計的性質が既知である場合には、この統計的性質と同じか良く似た統計的性質を持つ学習用入力信号を用いて、1 つの符号テーブルを作成しておくことで足りる。しかし、符号化対象の入力信号の統計的性質が未知である場合や変動するこ

50

となどが予想される場合には、複数の学習用入力信号から複数の符号テーブルを作成しておくことが望ましい。

【0064】

入力信号の統計的性質によっては、 s ビットで二進符号化された余り z のMSBと商 y のLSBとの間に強い相関関係が認められるような場合がある。このような場合、余り z のMSBを商 y に含めて符号化を行うことが良い。そこで、学習用入力信号から求められたRiceパラメータが $s(s-1)$ であるとき、 $s-1$ をRiceパラメータと看做して、0以上の整数に変換された学習用入力信号の各サンプルの商 y と余り z を求める。これにより、余り z の符号長は $s-1$ ビットに短縮される。一方、商 y の符号長は二進符号で1ビット増えるものの、例えば商 y にハフマン符号化を適用することにより、必ずしも商 y の符号長は増加しない。また、商 y の値はRiceパラメータが s の場合と比べて2倍になるから、規定整数の個数も2倍になりえる。このような場合であっても、規定整数に対応する符号の符号長が2倍になる訳ではない。余り z の符号長を短縮することで節約できた1ビット(0,1)を規定整数に対応する符号に付加することで2倍の個数の規定整数に対応できる。例えば表1の例において規定整数に対応する符号に1ビットを付加することで、表7(図27参照)に示すように、0から7までの8個の規定整数に対応できる(規定整数7に対応する符号が4ビット \times 2=8ビットではなく5ビットである。)。さらに、商 y の出現頻度(出現確率)によっては、Riceパラメータが s の場合と比べて2倍の個数の規定整数を規定する必然性はなく、表8の例(図28参照)のように2倍の個数よりも少ない個数の規定整数を規定することができ、この場合、表1の例と比較して、規定整数に対応する符号の符号長は4ビットで同じであるが、余り z の符号の符号長が1ビット短くなっている。

【0065】

<符号化処理>

次に、図17と図18を参照して、符号化装置1における符号化処理の流れを叙述的に説明する。

【0066】

[ステップSc1]

符号化装置1のバッファ部30は、入力信号をバッファして、入力信号をフレーム単位で出力する。

【0067】

[ステップSc2]

符号化装置1の分離パラメータ計算部110は、1フレームの入力信号の各サンプル x_i ($i=1, 2, \dots, N$)を用いて、式(9)または式(10)に従いRiceパラメータ $s = s_0$ または $s = s_0$ を求める。そして分離パラメータ計算部110は、求めた s または $s-1$ を分離パラメータ B として、分離パラメータ B を表す符号を出力する。なお、分離パラメータ B を $B = s - 1$ によって得る場合、 $s = 0$ のとき $B = 0$ とする。

【0068】

ここで、分離パラメータ B を表す符号は、例えば B の値の存在可能な範囲が $0 \leq B \leq 7$ である場合、一例として、3ビットの符号ないし正整数表現を用いて0から7までの値を表すようにしても良いし、分離パラメータ B の値の出現頻度に応じて、出現確率の高い値ほど短い一意の符号を割り当てるようにしてもよい。

後者の場合には、例えば2と3の出現確率が最も高い場合には以下のようなハフマン符号を符号化装置、復号化装置で共通に用いて符号化と復号化を行うことで、分離パラメータ B の値を一意に解決できる。

B=0: 100

B=1: 101

B=2: 00

B=3: 01

B=4: 110

10

20

30

40

50

B=5: 1110

B=6: 11110

B=7: 11111

また、日本国特許出願番号2009-056017（国際出願番号PCT/JP2010/053676）に開示された方法を用いて分離パラメータBを符号化しても良い。

【0069】

[ステップSc3]

符号化装置1の商余計算部120は、ステップSc2で決定された分離パラメータBに応じて、0以上の整数に変換した1フレームの入力信号の各サンプル x_i の商 y_i と余り z_i を求める。

商余計算部120は、まず、ステップSc2で決定された分離パラメータBが0と等しいか否かを判定する（ステップSc3-1）。B=0である場合には、商余計算部120は、0以上の整数に変換した1フレームの入力信号の各サンプル x_i について式(13)に従い商 y_i を求める（ステップSc3-2）。この場合、余り z_i は算出されない。B

1である場合には、商余計算部120は、0以上の整数に変換した1フレームの入力信号の各サンプル x_i について、式(12)に従い商 y_i を求め、式(14)に従い余り z_i を算出する（ステップSc3-3）。余り z_i は、サンプル x_i の正負を識別するための1ビットをMSBに付加してBビットの二進符号で表現される。

【0070】

[ステップSc4]

符号化装置1の商符号化部130は、ステップSc3で得られた商 y_i を記憶部70に記憶された符号テーブルを参照して符号化し、この商 y_i に対応する符号を出力する。この符号は実施形態1と同様の手順で求められる。

【0071】

商符号化部130は、複数の符号テーブルが存在する場合、分離パラメータBを与えたRiceパラメータsと同じ値のRiceパラメータを用いて作成された符号テーブルを用いて商 y_i の符号化を行う。複数の符号テーブルを用いて商 y_i の符号化を行った場合、得られた複数の符号のうち符号長が最小の符号を商符号化部130は出力する。この場合、商符号化部130は、出力する符号を得るために用いた符号テーブルを表す符号（符号テーブル特定符号）も出力する。記憶部に記憶された符号テーブルが一つだけの場合は、符号テーブル特定符号を出力する必要は無い。

【0072】

また、商符号化部130は、符号テーブルを用いた上述の符号化の他に、ステップSc3で得られた商 y_i に対して符号テーブルを用いない符号化を適用して符号を得てもよい。例えば、符号テーブルを用いない符号化方法として、ゴロム符号化、ゴロム ライス符号化、アルファ符号化などが例示される。この場合、符号テーブルを用いた符号化で得られた符号と符号テーブルを用いない符号化で得られた符号のうち符号長が最小の符号を商符号化部130は出力する。この場合、商符号化部130は、符号化方法を特定するための符号（符号化方法特定符号）と、符号化方法が符号テーブルを用いた符号化方法である場合には符号テーブル特定符号も出力する。

【0073】

合成部50は、出力された符号や符号テーブル特定符号などを多重化する。つまり、合成部50は、出力された、1フレームにつき一つ求められた分離パラメータBを表す符号、1フレームの各サンプルにつき商の符号と余りを表す二進符号を連結したビット列、必要に応じて、1フレームの各サンプルに対応する符号テーブル特定符号、符号化方法特定符号などをフレームごとにまとめて更にフレームの順番に従ってこれらを連結したビットストリームを出力する。ビットストリームは、図19に示す分離部40に入力される。分離部40は、ビットストリームを逆多重化して、符号化装置1から出力されたフレームごとの分離パラメータBを表す符号、フレームに含まれるサンプルごとの符号、フレームに含まれるサンプルごとの符号テーブル特定符号、符号化方法特定符号などを得る。

10

20

30

40

50

【 0 0 7 4 】

< 復号処理 >

次に、図 1 9 と図 2 0 を参照して、本発明の実施形態 3 の復号装置 2 における復号処理の流れを叙述的に説明する。

【 0 0 7 5 】

[ステップ S d 1]

復号装置 2 の商復号部 2 1 0 は、分離部 4 0 から送られたフレームに含まれる各サンプルの商の符号を復号する。この復号処理は、符号テーブル（分離部 4 0 によって符号テーブル特定符号が得られている場合には符号テーブル特定符号で特定される符号テーブルであり、符号テーブル特定符号が得られていない場合には既定の符号テーブルである。）を用いて、商の符号に拡張符号が含まれていなければ当該符号に対応する規定整数を出力し、商の符号に拡張符号が含まれていれば、拡張符号に対応する予め定められた符号化方法に対応する復号方法に従って当該符号から拡張符号を除いた部分を復号して得た整数を出力する処理である。出力された整数は商 y_i に相当する。

10

【 0 0 7 6 】

例えば符号テーブルが表 1 で示したものである場合、商 y_i の符号が 1 1 0 であるならば、この商 y_i の符号に拡張符号 1 1 1 1 が含まれていないので、符号 1 1 0 に対応する規定整数 2 を商 y_i として出力する。また、商 y_i の符号が 1 1 1 1 1 1 0 であるならば、この商 y_i の符号に拡張符号 1 1 1 1 が含まれているので、拡張符号 1 1 1 1 に対応するアルファ符号化 code(U,p-4) の復号方法に従って当該符号 1 1 1 1 1 1 0 から拡張符号 1 1 1 1 を除いた部分 1 1 0 を復号して得られる整数 2 に 4 を加えた 6 を商 y_i として出力する。ゴロム符号化、ゴロム-ライス符号化、アルファ符号化などに対応する各復号処理は、例えば上記非特許文献 3 に詳しいので、詳細な説明を略する。

20

【 0 0 7 7 】

商復号部 2 1 0 は、分離部 4 0 によって符号化方法特定符号が得られている場合には、符号化方法特定符号で特定される符号化方法に対応する復号方法で商の符号を復号する。符号化方法特定符号で特定される符号化方法が符号テーブルを用いる符号化方法である場合には、商復号部 2 1 0 は上述したような復号処理を行い、符号化方法特定符号で特定される符号化方法が符号テーブルを用いない符号化方法である場合には、商復号部 2 1 0 は符号化方法特定符号で特定される例えばゴロム符号化、ゴロム-ライス符号化、アルファ符号化などの符号化方法に対応する復号方法で商の符号を復号する。

30

【 0 0 7 8 】

[ステップ S d 2]

復号装置 2 の判定部 2 2 0 は、分離部 4 0 から送られた分離パラメータ B を表す符号から、分離パラメータ B が 0 に等しいか 1 以上であるかを判定する。B = 0 である場合、判定部 2 2 0 は、分離部 4 0 から送られたフレームに含まれる各サンプルの商の符号を第 1 復号部 2 3 0 へ送る制御を行う。B = 1 である場合、判定部 2 2 0 は、分離部 4 0 から送られたフレームに含まれる各サンプルの商の符号を第 2 復号部 2 4 0 へ送る制御を行う。

【 0 0 7 9 】

[ステップ S d 3]

復号装置 2 の第 1 復号部 2 3 0 は、(1) ステップ S d 1 で得られた整数 y_i の二進符号の L S B が 0 の場合、当該整数 y_i を 1 / 2 倍した整数を得、(2) ステップ S d 1 で得られた整数 y_i の二進符号の L S B が 1 の場合、当該整数 y_i について正負を表す符号を負に反転した負整数から 1 を減算した整数を 1 / 2 倍した整数を得る。この得られた整数が、B = 0 である場合における、復号されたサンプル x_i の値である。図中の記号 \gg は右シフト演算を表している。

40

【 0 0 8 0 】

[ステップ S d 4]

復号装置 2 の第 2 復号部 2 4 0 は、(1) 分離部 4 0 から送られたフレームに含まれる商 y_i に対応する余り z_i を表す二進符号の M S B が 1 であれば、ステップ S d 1 で得ら

50

れた整数 y_i を分離部 40 から送られた分離パラメータ B を用いて 2^{B-1} 倍した値の二進符号に、余り z_i を表す二進符号の下位 $B-1$ ビットを加算した整数を得、(2) 分離部 40 から送られたフレームに含まれる商 y_i に対応する余り z_i を表す二進符号の MSB が 0 であれば、ステップ S d 1 で得られた整数 y_i を分離部 40 から送られた分離パラメータ B を用いて 2^{B-1} 倍した値の二進符号に、余り z_i を表す二進符号の下位 $B-1$ ビット (または余り z_i を表す B ビットの二進符号) を加算して得られる整数について正負を表す符号を負に反転した負整数から 1 を減算した整数を得る。この得られた整数が、復号されたサンプル x_i の値である。図中の記号 \ll は左シフト演算を、記号 $\&$ はビット単位の論理積演算を、記号 $|$ はビット単位の論理和演算を表している。

【0081】

10

これまでの説明では、入力信号を、その各サンプルが正または負の整数で表されるものとして説明した。例えば、音声信号に適用する符号化方式によって入力信号の各サンプルが正の符号を持つ整数で表されているような場合、各サンプルの正負の符号を識別する必要が無い。このような場合には、分離パラメータ B を式 (10) を利用して求め、商 y_i を式 (15) または式 (16) で、余り z_i を式 (17) で求めればよい。

【数10】

$B \geq 1$ の場合

$$y_i = \left\lfloor \frac{x_i}{2^B} \right\rfloor \quad (15)$$

20

$B = 0$ の場合

$$y_i = x_i \quad (16)$$

$$z_i = x_i - \left\lfloor \frac{x_i}{2^B} \right\rfloor \cdot 2^B \quad (17)$$

【0082】

< 本発明の実施形態 4 の符号化の概要 >

30

実施形態 4 では、本発明の符号化を、G.711 であらかじめ符号化された音響信号のロスレス符号化に適用する場合の別の例を図 9 を用いて示す。

入力は G.711 符号化された音響信号で、サンプルあたり 8 ビットの符号により構成される符号列であり、以下では G.711 符号列と呼ぶ。入力された G.711 符号列は、符号化装置 800 E にて、指定されたフレーム処理単位ごとに符号化処理される。

フレーム処理単位は、40 サンプル、80 サンプル、160 サンプル、240 サンプル、320 サンプルの 5 通りのうちのいずれかで、符号化時に外部から指定されることとする。

【0083】

入力された G.711 符号列はバッファ部 802 E で、指定されたフレーム処理単位 (たとえば 40 サンプル) ごとに分割され、変換部 804 E に渡される。また、バッファ部 802 E では指定されたフレーム処理単位を識別するための符号を、フレーム処理単位指定符号として出力する。

40

【0084】

変換部 804 E では、フレーム処理単位毎の G.711 符号列中の各符号を、対応する PCM 値に変換し、フレーム処理単位毎の PCM 値列を出力する。

【0085】

予測分析部 806 E では、変換部 804 E が出力した PCM 値列に対して予測分析を行い、予測係数を決定する。(このとき、予測分析部 806 E では線形予測分析等用いることができるが、必ずしも線形予測分析に限る必要は無く、何らかの予測を用いて PCM 値列に

50

対する予測係数を得ればよい。)

【0086】

予測係数量子化部808Eでは、予測分析部806Eで決定された予測係数を量子化して、量子化予測係数を得ると同時に、量子化予測係数に対応する量子化予測係数符号を出力する。

【0087】

予測部810Eでは、量子化予測係数と過去のPCM値列を用いて、フレーム処理単位毎の現在のサンプルのPCM値の列を推定する。

【0088】

逆変換・変形部812Eでは、予測部810Eで推定されたPCM値の列の各PCM値をG.711符号化し、符号の表す値の大小関係を保存した形の、例えば-128から+127までの2の補数表現された8ビットの値(推定サンプル値)に変換する。ここで、たとえば日本国特許出願番号2007-319805(国際出願番号PCT/JP2008/072513)に記載されているような変形を行うこととしても良い。

【0089】

例えば、次のような変形を行う。特定の範囲に含まれる番号であって、G.711符号列 = { (1), (2), ..., (N) } 内に発生していない番号があるかを確認する。Nは1フレームのサンプル数である。特定の範囲とは、例えば、正の絶対値が最小の値を示す番号と負の絶対値が最小の値を示す番号である。具体的には、ITU-T G.711の μ 則の場合であれば“+0”と“-0”、A則の場合であれば“+8”と“-8”である。発生していない番号があると分かった場合には、当該発生していない番号が示す元の信号の大きさを除いて、元の信号の大きさを示す番号を付加しなおした番号に、G.711符号列の各番号を置き換えた番号の列 $T(\) = \{ T(\ (1)), T(\ (2)), \dots, T(\ (N)) \}$ を出力する。例えば、ITU-T G.711の μ 則の場合を考える。 μ 則では8ビットで“-127”から“+127”の番号を示しているが、“0”を示す番号には“+0”と“-0”の2つがある。そして、元の信号と線形な関係の値との関係では、“-127”は「-8031」、 “+127”は「+8031」、 “+0”と“-0”は「0」を示している。ここでは、“ ”は元の信号の大きさ(大小関係)を示す番号を示しており、「 」は元の信号と線形な関係の信号の振幅を示している。このように、“+0”と“-0”とは重複した番号なので、符号化装置によってはどちらか一方だけを出力する装置もある。そこで、例えば特定の範囲を“+0”と“-0”にする。そして、“-0”が発生しないのであれば、負の番号は1つずつずらし、“-0”が「-1」を示す番号、“-126”が「-8031」を示す番号として用いる。“+0”が発生しないのであれば、正の番号は1つずつずらし、“+0”が「+1」を示す番号とすればよい。また、“+0”も“-0”も存在しないのであれば、負の番号も正の番号も1つずつずらし、“-0”が「-1」を示す番号、“+0”が「+1」を示す番号とすればよい。

【0090】

変形部814Eでは、フレーム処理単位毎のG.711符号列中の各符号を、符号の表す値の大小関係を保存した形の、例えば-128から+127までの2の補数表現された8ビットの値(変形入力サンプル値)に変換する。ここでも、たとえば日本国特許出願番号2007-319805(国際出願番号PCT/JP2008/072513)に記載されているような変形を行うこととしても良く、その一例は上述したとおりである。

【0091】

誤差計算部816Eでは、逆変換・変形部812Eから得られた8ビット表現された推定サンプル値と、変形部814Eから得られた8ビット表現された変形入力サンプル値との差分を求めることで、変形入力サンプル値と推定サンプル値との誤差を計算する。この誤差の系列が本発明の符号化方法に入力される整数値系列 x_i である。

【0092】

分離パラメータ計算部818Eは、誤差の系列をフレーム処理単位のサンプル数だけ入力として誤差計算部816Eから受け取り、フレーム処理単位ごとに分離パラメータBを決

10

20

30

40

50

定する。

また、分離パラメータ計算部 8 1 8 E は分離パラメータ B に対応する符号を分離パラメータ符号として出力する。実施形態 3 で説明したように、たとえば、分離パラメータ B の出現頻度に応じて可変長符号化したハフマンテーブルをあらかじめ決めておき、そのハフマンテーブルを用いて分離パラメータ B の符号を得る。

【 0 0 9 3 】

商余計算部 8 2 0 E は、分離パラメータ計算部 8 1 8 E で決定した分離パラメータ B を用いて、誤差系列 x を商系列 y と余系列 z に分離する。また、商余計算部 8 2 0 E は余系列に含まれる各余りを B ビットで符号化して得られる余符号列を出力する。

【 0 0 9 4 】

誤差系列 x に含まれるサンプル x_i ($i = 1, 2, \dots, N$; N は 1 フレームに含まれるサンプル数) について、0 以上の整数に変換したこのサンプル x_i の分離パラメータ B を用いた商 y_i および余 z_i を実施形態 3 の場合と同様に、式 (12)、式 (13) で求める。式 (12) により商 y_i を求める場合は、余りを表す符号を、余りを表す B - 1 ビットの二進符号にサンプル x_i の正負を識別するための 1 ビットの符号が付加された B ビットの固定長を持つ符号とする。式 (13) により商 y_i を求める場合は、商 y_i を表す符号の LSB (Least Significant Bit) によってサンプル y_i の正負が識別される。ここで、分離パラメータ B を用いた商 y_i および余 z_i の分離方法は、一意に分離、合成が可能であれば別の分離方法を用いることとしても良い。

【 0 0 9 5 】

本実施形態 4 では、フレーム処理単位に応じて符号テーブルの個数をフレーム処理単位が 4 0 サンプルの場合には 1 種類、フレーム処理単位が 8 0 サンプル以上の場合には 4 種類の中から 1 つを選択して用いるように切り替える。

これは、フレーム処理単位として、比較的小さな 4 0 サンプルが指定された場合には、誤差系列の分散が異なる 4 種類の符号テーブルの中から 1 つを選択するために、符号化テーブル指定符号に情報量を割り当てても、符号化対象となる誤差係数のサンプル数が少ない (4 0 サンプル) ために、増加した符号テーブル指定符号分の情報量を補うほど誤差系列に対応する符号列の符号量の減少が見込めないため、結果的に符号テーブルを複数用いるメリットが無いためである。

これに対してフレーム処理単位が 8 0 サンプル以上の場合には、符号テーブル指定符号を用いて 4 種類の符号テーブルから誤差系列の統計的な性質に最も適した符号テーブルを選択することで、符号テーブル指定符号に用いる符号量を考慮したとしても、誤差系列に対応する符号列の符号量を減少させることが出来ることから、結果として総符号量をより少なくすることが出来る。

図 1 0 に商余計算部 8 2 0 E での計算によって得られた商系列に含まれる値の分布例を示す。

振幅の小さな値は比較的出現頻度が高いが、破線で示すラプラス分布とは異なる場合もあるため、複数の符号テーブルの中で、当該フレーム内の商系列に含まれる値の出現頻度分布に最も近い頻度分布をあらわしているものを選択して用いる。また、振幅の大きな値の出現頻度がラプラス分布と比較して大きくなっていると、アルファ符号では非常に長い符号が必要となる場合がある。

図 1 1 に、実施形態 4 でフレーム処理単位が 4 0 サンプルの場合に用いる符号テーブルの例、およびフレーム処理単位 8 0 以上の場合の符号テーブルの例を示す。フレームサイズおよび分割パラメータ B によって符号テーブルおよび拡張符号に対応する符号化方式を切り替える。

【 0 0 9 6 】

符号テーブル選択部 8 2 2 E では、商系列を入力として、出力される符号量が最小となるように記憶部 8 2 4 E に記憶された符号テーブル 8 2 4 1 を選択し、選択された符号テーブルに対応する符号テーブル指定符号を出力する。フレーム処理単位が 8 0 サンプル以上の場合には、テーブル 0 からテーブル 3 に示した 4 種類の符号テーブル (図 1 1 参照) の

10

20

30

40

50

中から、当該フレームの商符号列の総符号量が最も少なくなる符号テーブルを選択し対応する符号テーブル指定符号を出力する。フレーム処理単位が40サンプルの場合には、符号テーブルは1種類に限定されているため、符号テーブル指定符号は出力されない。

【0097】

商符号化部826Eは、符号テーブル選択部822Eで選択された符号テーブルおよび、フレーム処理単位と分離パラメータBによって一意に決定された、拡張符号に対応する符号化方式を用いて商系列を符号化し、商符号列を出力する。

【0098】

このとき、分離パラメータBの値に応じて $B = 0$ の場合には、図11の上段のテーブル0からテーブル3に示すように、フレーム処理単位にかかわらず、拡張符号に対応した符号化方式として、アルファ符号 $code(U, p1-maxCode)$ を用いる。フレーム処理単位が40サンプルで $B = 0$ の場合に出力される商符号列に含まれる商符号の例を図12に示す。

$B > 0$ の場合には、図11の下段のテーブル0からテーブル3に示すように、フレーム処理単位にかかわらず、拡張符号に対応した符号化方式として、ライスパラメータ $s=1$ のゴロムライス符号 $code(R, 1, p2-maxCode)$ を用いる。フレーム処理単位が40サンプルで $B > 0$ の場合に出力される商符号列に含まれる商符号の例を図13に示す。

分離パラメータBの計算式より、誤差系列の振幅が小さい場合に分離パラメータ $B = 0$ となり、誤差系列の振幅が大きい場合に分離パラメータ $B > 0$ となる。

誤差系列の振幅が小さい場合、すなわち分離パラメータ $B = 0$ の場合には、ラプラス分布に近いが、誤差系列の振幅が大きい場合、すなわち分離パラメータ $B > 0$ の場合には必ずしもラプラス分布に従わないことが実験的に観測されている。

観測によれば、分離パラメータ $B > 0$ となるような場合には、ラプラス分布よりも相対的に振幅の大きな商の値の出現確率が高いことがわかっている。

本実施形態4では、このことを考慮して、分離パラメータ $B > 0$ の場合には、誤差符号を分離パラメータBにより商と余りに分離し、その商のうち、値があらかじめ符号テーブルごとに定められた値 $maxCode$ 以上のものに対して、さらにライスパラメータ $s=1$ のゴロムライス符号化を用いて符号化する。

一方で、分離パラメータ $B = 0$ の場合には、商をアルファ符号 $code(U, p-maxCode)$ で符号化する。ここで、誤差系列を分離パラメータBにより商と余りに分離する処理は、 $B = 0$ では余りは存在しないので商系列のみが出力される。

このように、分離パラメータBの値と商系列の各値の出現確率の間には相関関係が存在しているので、分離パラメータBの値に応じて符号テーブルおよび拡張符号に対応する符号化方式の一方もしくは両方を切り替えることで符号化効率を向上させることができる。同様にフレーム処理単位に応じて符号テーブルおよび拡張符号に対応する符号化方式を切り替える。

【0099】

合成部830Eでは、上記各処理部より出力された、フレーム処理単位指定符号、量子化予測係数符号、分離パラメータ符号、符号テーブル指定符号、商符号列、余符号列をフレーム毎にまとめて符号系列として出力する。

【0100】

<本発明の実施形態4の復号化の概要>

本発明の実施形態4の復号処理について図14を用いて説明する。

復号装置850Dの入力は、前記実施形態4の符号化により生成した符号系列である。

【0101】

分離部860Dでは符号系列を入力として受け取り、フレーム処理単位指定符号、量子化予測係数符号、分離パラメータ符号、符号テーブル指定符号、商符号列、余符号列、に分離して各処理部に送る。

【0102】

商復号部870Dでは、符号テーブル指定符号、および、フレーム処理単位指定符号により当該フレームの商符号列の復号に用いる符号テーブル8721(記憶部872Dに記

10

20

30

40

50

憶されている)を選択する。また、商復号部 870D は、前記選択した符号テーブルを用いて、入力された商符号列の各商符号に対応する整数値の系列を商系列として出力する。

商系列は、商余合成部 874D に送られる。

【0103】

商余合成部 874D では、分離パラメータ符号を復号して分離パラメータ B を得て、余符号列を復号して得られる各余り z_i と、商系列に含まれる商 y_i とを用いて次式の計算により誤差系列に含まれる各サンプル x_i を得る。ここで、 $y\%2$ は y を 2 で除算した余りを表す。得られた誤差系列は、誤差加算部 876D に送られる。

【数11】

B=0 の場合

$$x_i = \begin{cases} y_i/2 & (y\%2=0) \\ -y_i/2 & (y\%2>0) \end{cases}$$

B>=1 の場合

$$x_i = \begin{cases} y_i \times 2^{B-1} + z_i & (z < 2^{B-1}) \\ -y_i \times 2^{B-1} - (z_i - 2^{B-1}) & (z \geq 2^{B-1}) \end{cases}$$

【0104】

量子化予測係数復号部 878D では、分離部 860D から受け取った量子化予測係数符号を復号して、量子化予測係数を得る。

【0105】

予測部 880D では、変換部 884D により復号された過去の G.711 符号列に対応する 8 ビットの整数値を、対応する PCM 値に変換した系列を保持しており、この過去の PCM 値列と、量子化予測係数を用いて、フレーム処理単位毎の当該 PCM 値の列を推定する。

逆変換・変形部 882D では、予測部 880D で推定された PCM 値の列の各 PCM 値を、G.711 符号化し、符号の表す値の大小関係を保存した形の、例えば -128 から +127 までの 2 の補数表現された 8 ビットの値 (推定サンプル値) に変換する。ここで、符号化処理時に、日本国特許出願番号 2007-319805 (国際出願番号 PCT/JP2008/072513) に記載されているような変形を行った場合には、復号処理時にも対応する処理を行う。

【0106】

誤差加算部 876D では、逆変換・変形部 882D から得られた推定サンプル値と誤差系列中の当該誤差値を加算して、符号化処理時の「変形入力サンプル値」に相当する、元の G.711 符号に対応し、符号の表す値の大小関係を保存した形の、例えば -128 から +127 までの 2 の補数表現された 8 ビットの値 (復号変形入力サンプル値) を得る。

誤差加算部 876D では、さらに、当該フレーム処理単位内の全てのサンプルについて同様の処理を行い、復号変形入力サンプル値系列を得る。

【0107】

逆変形部 886D では前記復号された復号変形入力サンプル値系列から、元の G.711 符号列を復元する。

【0108】

< 商復号部での復号処理の詳細 >

商復号部 870D において、商符号列から商系列を得るに際して、復号テーブルを利用することにしても良い。本実施形態では、商符号列の復号に復号テーブルを用いる場合の好適な実施の形態について説明する。

例えば、フレーム処理単位の 80 サンプル以上の場合の符号テーブル 0 に対応する復号テーブルの例を図 15 に示す。復号処理手順の例を図 16 に示す。

【0109】

まず、商符号列のうち、未処理の商符号の先頭のビットを処理開始点とする (ステップ S850)。未処理の商符号が存在するか否か判定する (ステップ S852)。存在しな

10

20

30

40

50

いならば処理を終了する。存在すれば、符号テーブル指定符号に基づき、当該フレーム処理単位内（フレーム内）の商符号の復号に用いる復号テーブルvalue[],Len[]を選択する（ステップS854）。そして、フレーム内の商符号列の復号が完了したかを判定する（ステップS856）。完了した場合は、ステップS852の処理を行う。完了していない場合、商符号列のうち、未処理の商符号の先頭のビットを処理開始点とする（ステップS858）。商符号列から、1つの商符号に許容されている最長符号長に相当するビット数（実施例では6ビット）を読み込み0以上の整数値として変数vに設定する（ステップS860）。変数vの値で復号テーブルvalue[]を参照し商の値を $q = \text{value}[v]$ とする（ステップS862）。商qの値で復号テーブルLen[]を参照し、当該符号のビット長 $I = \text{Len}[q]$ を得る（ステップS864）。商符号列の処理地点を当該符号長に相当するIビットだけ進め、未処理の商符号の先頭に移動する（ステップS866）。規定整数に対応する商符号かどうか確認（ $q - \text{codeMax} < 0$ ）する（ステップS868）。規定整数であれば、qを商の値として出力する（ステップS870）。規定整数でなければ、拡張符号から拡張符号に対応する符号化方式を判別し、判別された符号化方式で拡張符号に続く符号を復号して得られる整数値を商の値として出力する（ステップS872）。当該符号に対応する符号も含めて読み込んだビット数だけ読み込み位置を移動し、未処理の商符号の先頭のビットを処理開始点とする（ステップS874）。ステップS870およびS874の後に、ステップS856の処理を行う。

10

【0110】

つまり、商復号部870Dでは、符号系列から符号テーブル指定符号を読み込み、符号テーブル指定符号に対応する復号テーブルを選択する。この復号テーブルには、符号テーブルに対応した値が登録されている。

20

【0111】

商符号は、先頭から連続するビット列として入力され、可変ビット長の符号である。商復号部は、先頭より最大符号長に相当するビット数だけ整数値として読み込む。この例では、符号テーブルに登録された符号のうち、最長の符号長は6ビットであるので、6ビットを読み込んで、0以上の整数とみなして変数vに設定する。ここで、もし読むべき符号が存在しない場合には、最長符号長に達するまでLSB側に値0のビットを付与して、最長符号長にしてから配列を参照すればよい。

【0112】

次に、符号テーブル指定符号により選択された符号テーブルに対応する復号テーブルであるvalue[]テーブルを用いてこの6ビットの商符号に対応する値qを復号する。すなわち、配列の添え字に前記設定したvを用いて $q = \text{value}[v]$ とすれば復号後の値qが求まる。

30

【0113】

次に、実際の当該符号の符号長Iを求める。符号長は $I = \text{len}[v]$ もしくは $I = \text{Len}[q]$ として容易に求めることができる。得られたIビット分だけ商符号の処理位置を移動することで、未処理の商符号の先頭位置に移動することが出来る。

【0114】

このように、可変符号長の符号の復号にもかかわらず、最長符号長のビットを無条件に読み込んで、その値を元に配列を参照するだけで、複雑な復号処理を行うことなく高速に復号結果を得ることが出来る。

40

同様に、当該符号長も配列を参照するだけで得ることが出来る。

また、符号テーブルの設計に際して、最長符号長のサイズを一定にすることで、復号テーブルのデータ保持に必要なメモリサイズを制限することが出来る。

本実施例では、図15の符号テーブルの設計に際して、各符号テーブルに登録される符号のうち、最長の符号長が6ビットに収まるように制約することで、復号テーブルのサイズを制限している。

【0115】

50

次に、得られた復号値 q が、 $codeMax$ よりも大きいかどうかを判定することで、当該商符号が、規定整数に対応した符号であるかどうかを確認する。 $(q-codeMax)<0$ であれば、当該商符号は規定整数に対応する符号であり、 $(q-codeMax)\geq 0$ であれば当該商符号は拡張符号である。

【0116】

当該商符号が規定整数に対応する符号である場合には、得られた q を商の値として出力し、処理開始点を当該符号長分だけ移動させる。当該符号長 I は、前述のように $I=Len[q]$ として求めることができる。

【0117】

当該商符号が拡張符号に対応する符号である場合には、得られた q に対応する拡張符号を特定する。(図の例の場合は $q=7$ のとき拡張符号 $p1$)

また、同様に拡張符号の符号長 I だけ符号読み込み位置を移動させることで、処理開始点を未処理の商符号の先頭に移動させる。当該符号長 I は、前述のように $I=Len[q]$ として求めることができる。

続いて、拡張符号に対応する復号方式を用いて、拡張符号に続く符号を復号することで、商 q を得る。

当該拡張符号に対応する符号化符号も含めて、読み込んだビット数だけ読み込み位置を移動し、未処理の商符号の先頭ビットを処理開始点とする。

【0118】

上記の処理を当該フレームのサンプル数に相当する数だけ商符号列を復号できた時点で、当該フレーム分の処理を終了する。

【0119】

以上の各実施形態の他、本発明である符号化装置・方法、復号装置・方法は上述の実施形態に限定されるものではなく、本発明の趣旨を逸脱しない範囲で適宜変更が可能である。また、上述の説明に現れる技術的特徴を互いに矛盾しない範囲で自由に組み合わせて実施できる。

【0120】

また、上記符号化装置、復号装置における処理機能をコンピュータによって実現する場合、符号化装置、復号装置が有すべき機能の処理内容はプログラムによって記述される。そして、このプログラムをコンピュータで実行することにより、上記符号化装置、復号装置における処理機能がコンピュータ上で実現される。

【0121】

この処理内容を記述したプログラムは、コンピュータで読み取り可能な記録媒体に記録しておくことができる。コンピュータで読み取り可能な記録媒体としては、例えば、磁気記録装置、光ディスク、光磁気記録媒体、半導体メモリ等のようなものでもよい。具体的には、例えば、磁気記録装置として、ハードディスク装置、フレキシブルディスク、磁気テープ等を、光ディスクとして、DVD (Digital Versatile Disc)、DVD-RAM (Random Access Memory)、CD-ROM (Compact Disc Read Only Memory)、CD-R (Recordable) / RW (ReWritable) 等を、光磁気記録媒体として、MO (Magneto-Optical disc) 等を、半導体メモリとしてEEPROM (Electrically Erasable and Programmable-Read Only Memory) 等を用いることができる。

【0122】

また、このプログラムの流通は、例えば、そのプログラムを記録したDVD、CD-ROM等の可搬型記録媒体を販売、譲渡、貸与等することによって行う。さらに、このプログラムをサーバコンピュータの記憶装置に格納しておき、ネットワークを介して、サーバコンピュータから他のコンピュータにそのプログラムを転送することにより、このプログラムを流通させる構成としてもよい。

【0123】

このようなプログラムを実行するコンピュータは、例えば、まず、可搬型記録媒体に記録されたプログラムもしくはサーバコンピュータから転送されたプログラムを、一旦、自

10

20

30

40

50

己の記憶装置に格納する。そして、処理の実行時、このコンピュータは、自己の記録媒体に格納されたプログラムを読み取り、読み取ったプログラムに従った処理を実行する。また、このプログラムの別の実行形態として、コンピュータが可搬型記録媒体から直接プログラムを読み取り、そのプログラムに従った処理を実行することとしてもよく、さらに、このコンピュータにサーバコンピュータからプログラムが転送されるたびに、逐次、受け取ったプログラムに従った処理を実行することとしてもよい。また、サーバコンピュータから、このコンピュータへのプログラムの転送は行わず、その実行指示と結果取得のみによって処理機能を実現する、いわゆるASP (Application Service Provider) 型のサービスによって、上述の処理を実行する構成としてもよい。なお、本形態におけるプログラムには、電子計算機による処理の用に供する情報であってプログラムに準ずるもの(コンピュータに対する直接の指令ではないがコンピュータの処理を規定する性質を有するデータ等)を含むものとする。

10

【0124】

また、この形態では、コンピュータ上で所定のプログラムを実行させることにより、符号化装置、復号装置を構成することとしたが、これらの処理内容の少なくとも一部をハードウェア的に実現することとしてもよい。

【図1】

【図2】

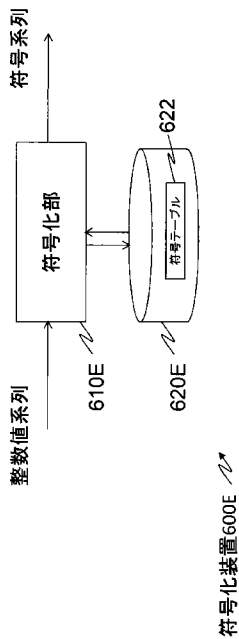


図1

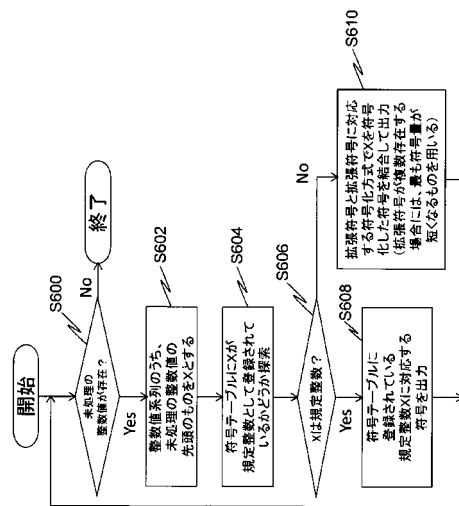


図2

【 図 3 】

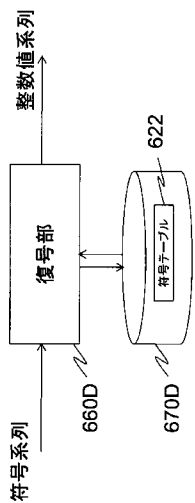


図3

復号装置650D

【 図 4 】

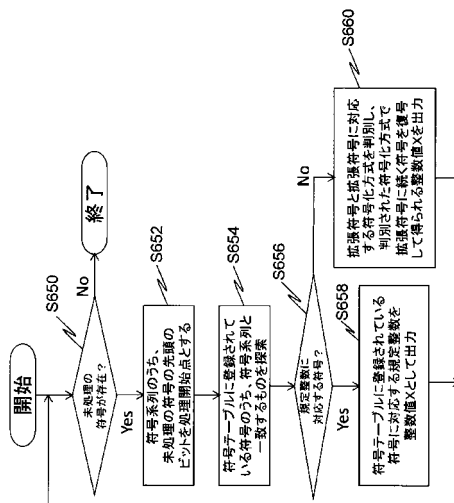


図4

【 図 5 】

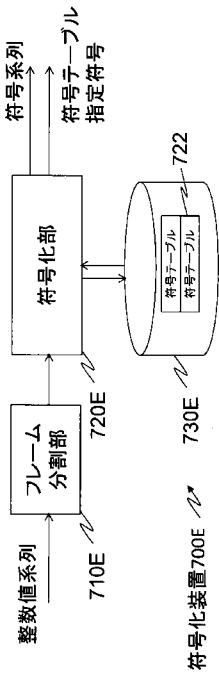


図5

符号化装置700E

【 図 6 】

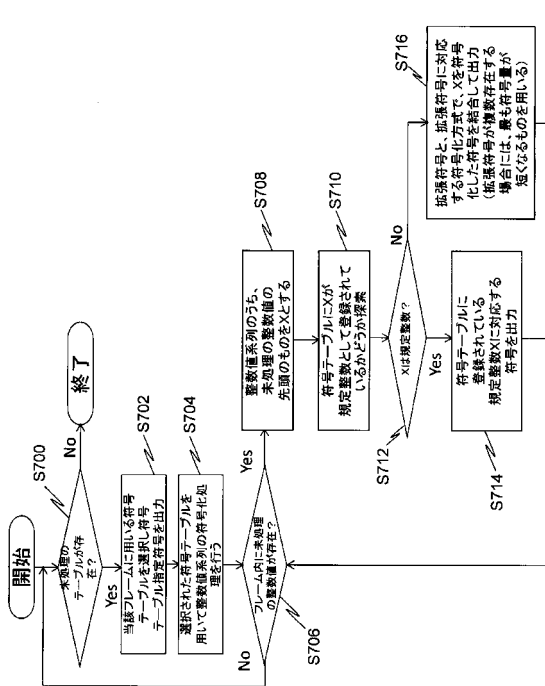


図6

【 図 7 】

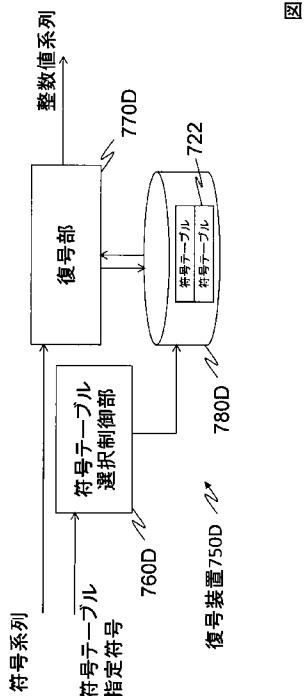


図7

【 図 8 】

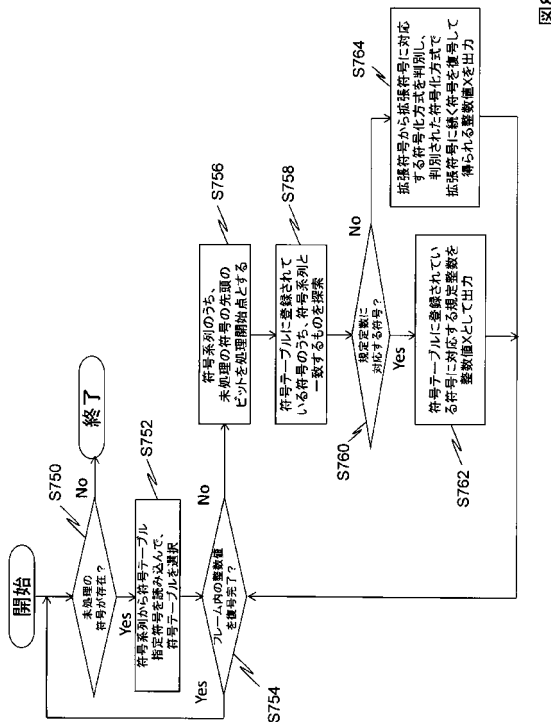


図8

【 図 9 】

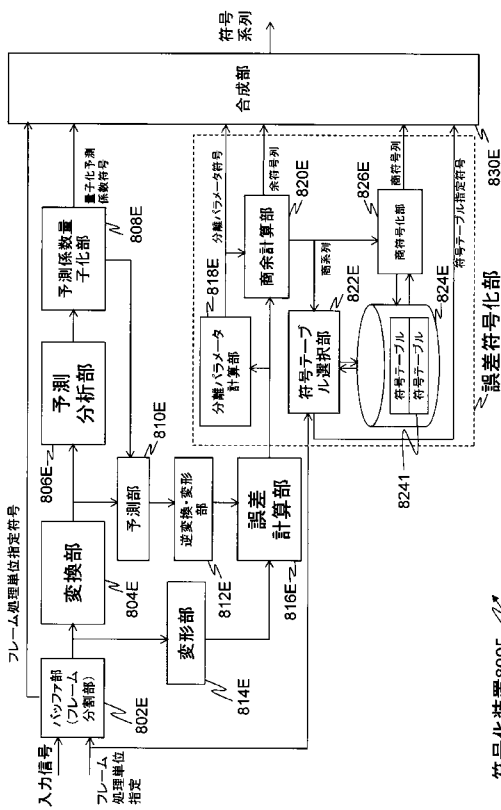


図9

【 図 10 】

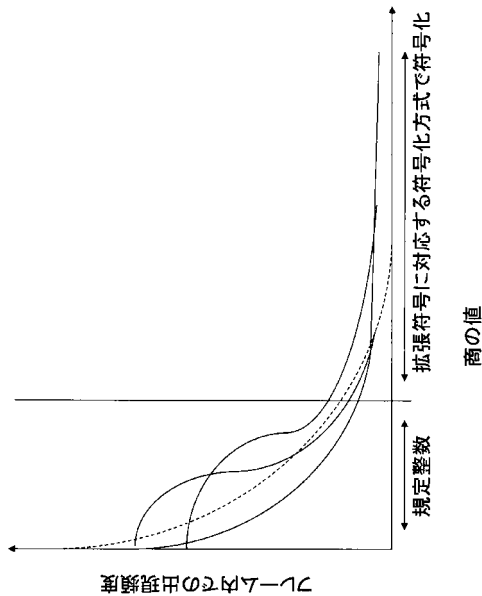


図10

【 図 15 】

maxCode: 7	value[0] = 6	0000000	0000000	value[32] = 1	1000000	len[0] = 6
k: code	value[1] = 7	0000001	0000001	value[33] = 1	1000001	len[1] = 6
0: 01	value[2] = 5	0000010	0000010	value[34] = 1	1000010	len[32] = 2
1: 10	value[3] = 5	0000011	0000011	value[35] = 1	1000011	len[33] = 2
2: 11	value[4] = 4	0000100	0000100	value[36] = 1	1000100	len[34] = 2
3: 001	value[5] = 4	0000101	0000101	value[37] = 1	1000101	len[35] = 2
4: 0001	value[6] = 4	0000110	0000110	value[38] = 1	1000110	len[36] = 2
5: 00001	value[7] = 4	0000111	0000111	value[39] = 1	1000111	len[37] = 2
6: 000000	value[8] = 3	0001000	0001000	value[40] = 1	1001000	len[38] = 2
pl: 000001	value[9] = 3	0001001	0001001	value[41] = 1	1001001	len[39] = 2
	value[10] = 3	0001010	0001010	value[42] = 1	1001010	len[40] = 2
	value[11] = 3	0001011	0001011	value[43] = 1	1001011	len[41] = 2
	value[12] = 3	0001100	0001100	value[44] = 1	1001100	len[42] = 2
	value[13] = 3	0001101	0001101	value[45] = 1	1001101	len[43] = 2
	value[14] = 3	0001110	0001110	value[46] = 1	1001110	len[44] = 2
	value[15] = 3	0001111	0001111	value[47] = 1	1001111	len[45] = 2
	value[16] = 0	0100000	0100000	value[48] = 2	1100000	len[46] = 2
	value[17] = 0	0100001	0100001	value[49] = 2	1100001	len[47] = 2
	value[18] = 0	0100010	0100010	value[50] = 2	1100010	len[48] = 2
Len[0] = 2	value[19] = 0	0100011	0100011	value[51] = 2	1100011	len[49] = 2
Len[1] = 2	value[20] = 0	0101000	0101000	value[52] = 2	1101000	len[50] = 2
Len[2] = 3	value[21] = 0	0101001	0101001	value[53] = 2	1101001	len[51] = 2
Len[3] = 3	value[22] = 0	0101010	0101010	value[54] = 2	1101010	len[52] = 2
Len[4] = 4	value[23] = 0	0101011	0101011	value[55] = 2	1101011	len[53] = 2
Len[5] = 5	value[24] = 0	0110000	0110000	value[56] = 2	1110000	len[54] = 2
Len[6] = 6	value[25] = 0	0110001	0110001	value[57] = 2	1110001	len[55] = 2
Len[7] = 6	value[26] = 0	0110100	0110100	value[58] = 2	1110100	len[56] = 2
	value[27] = 0	0110101	0110101	value[59] = 2	1110101	len[57] = 2
	value[28] = 0	0111000	0111000	value[60] = 2	1111000	len[58] = 2
	value[29] = 0	0111001	0111001	value[61] = 2	1111001	len[59] = 2
	value[30] = 0	0111100	0111100	value[62] = 2	1111100	len[60] = 2
	value[31] = 0	0111101	0111101	value[63] = 2	1111101	len[61] = 2
	value[32] = 2	0111110	0111110	value[64] = 2	1111110	len[62] = 2
	value[33] = 2	0111111	0111111	value[65] = 2	1111111	len[63] = 2

【 図 17 】

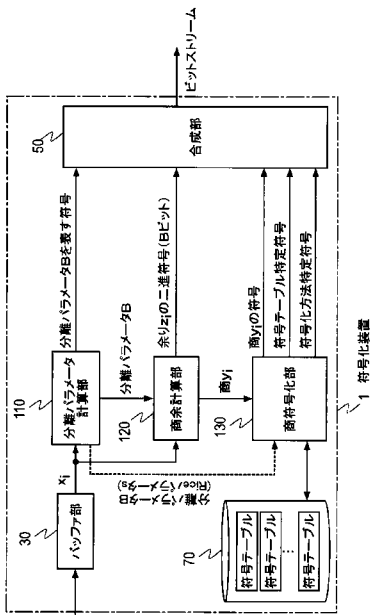


図17

【 図 16 】

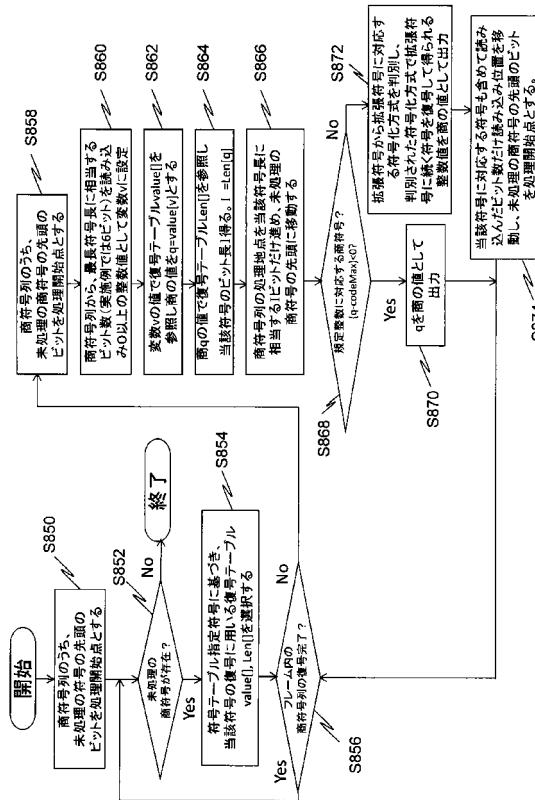


図 16

【 図 18 】

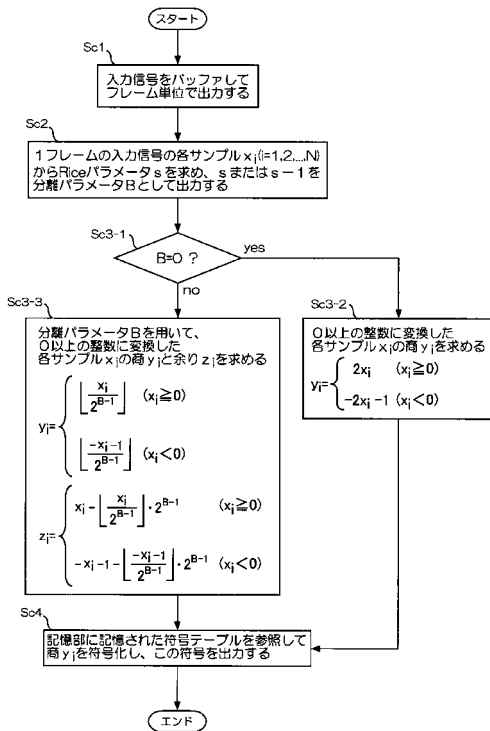


図18

【図19】

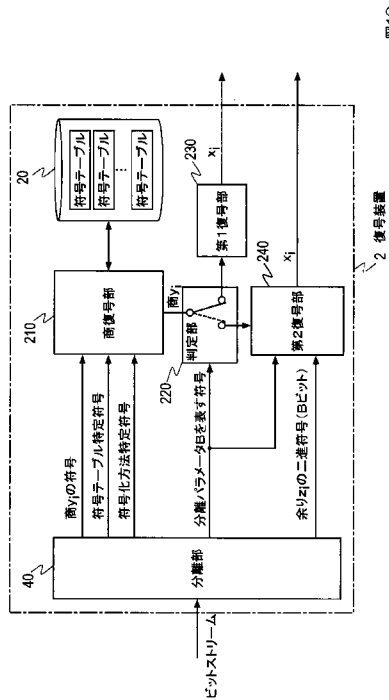


図19

【図20】

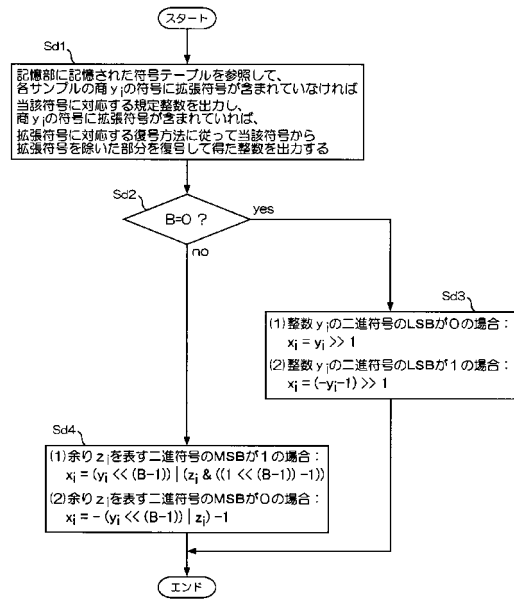


図20

【図21】

●表1

0:0
1:10
2:110
3:1110
p:1111 code(U,p-4)

図21

【図22】

●表2

0:0
1:10
2:110
3:1110
p:11110 code(U,p-4)
11111 code(F,G,s-1) code(R,s,p-4)

図22

【 図 2 3 】

●表3

0:0
1:10
2:110
3:1110
p:111100 code(U,p-4)
111101 code(R,1,p-4)
111110 code(R,2,p-4)
111111 code(F,g,s-3) code(R,s,p-4)

図23

【 図 2 4 】

●表4

0:0
1:10
2:110
3:1110
p:111110 code(U,p-4;4 ≤ p ≤ 6)
111110 code(U,p-12;15 ≤ p)
111111 code(F,h,p-7;7 ≤ p ≤ 14)

図24

【 図 2 5 】

●表5

0:0
1:10
2:110
3:1110
4:111100
p:111101 code(U,p-5)
111110 code(F,3,p-5)
111111 code(F,4,p-5)

図25

【 図 2 6 】

●表6

0:0
1:10
2:11110
4:111111
p:11110 code(U,p-3;p=3)
11110 code(U,p-4;5 ≤ p)
110 code(F,3,p-4;5 ≤ p)

図26

【 図 27 】

●表7

0:00
1:01
2:100
3:101
4:1100
5:1101
6:11100
7:11101
p:11111 code(U,p-4)

図27

【 図 28 】

●表8

0:00
1:01
2:100
3:101
4:1100
5:1101
p:11111 code(U,p-4)

図28

フロントページの続き

(72)発明者 守谷 健弘

東京都千代田区大手町二丁目3番1号 日本電信電話株式会社内

審査官 北村 智彦

(56)参考文献 特表2008-514142(JP,A)

特開2004-104159(JP,A)

Liebchen,t.,Reznik,Y.A., MPEG-4 ALS:an Emerging Standard for Lossless Audio Coding, Proceedings of the Data Compression Conference(DCC'04), 2004年 3月, pp.439-448

(58)調査した分野(Int.Cl., DB名)

H03M 3/00 - 11/00

G10L 19/00

H04N 1/41