(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2010-78965
(P2010-78965A)
(43)公開日　平成22年4月8日(2010.4.8)

(51)Int.Cl.		F I		テーマコード（参考）
G10L 19/02	(2006.01)	G10L 19/02	160A	
G10L 19/00	(2006.01)	G10L 19/00	100	
		G10L 19/02	144	

審査請求 未請求　請求項の数 10　OL　（全 25 頁）

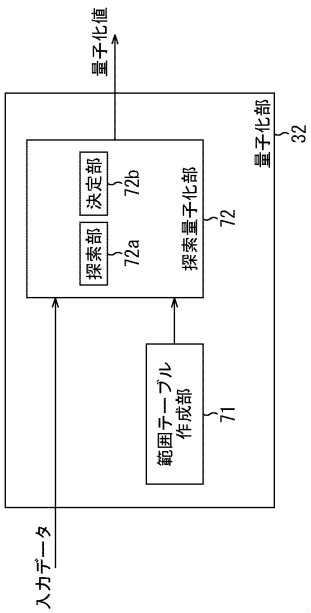
(21)出願番号　特願2008-247683 (P2008-247683)
(22)出願日　平成20年9月26日 (2008.9.26)

(71)出願人　000002185
ソニー株式会社
東京都港区港南１丁目７番１号
(74)代理人　100082131
弁理士　稲本　義雄
(74)代理人　100121131
弁理士　西川　孝
(72)発明者　茂木　幸彦
東京都港区港南１丁目７番１号　ソニー株式会社内
(72)発明者　鎌田　征人
東京都港区港南１丁目７番１号　ソニー株式会社内

(54)【発明の名称】演算装置および方法、量子化装置および方法、並びにプログラム

(57)【要約】
【課題】より効率良く非線形な演算を行う。
【解決手段】範囲テーブル作成部７２は、入力データＸに対応する量子化前の値Ｙに対して非線形な演算を施して得られる演算結果を量子化した量子化値Ｚと、量子化値Ｚをとりうる量子化前の値Ｙの範囲とを対応付けた範囲テーブルを作成し、探索量子化部７３は、入力データが入力されたとき、範囲テーブルにおいて、入力された入力データＸに対応する量子化前の値Ｙが含まれる範囲を探索し、探索された範囲に対応する量子化値Ｚを出力する。本発明は、例えば、量子化処理を行うオーディオ符号化装置に適用することができる。
【選択図】図２

図2



【特許請求の範囲】
【請求項１】
入力値に対して非線形な演算を施して得られる演算結果を離散化した離散値と、前記離散値をとりうる前記入力値の範囲とを対応付けた範囲テーブルを作成する範囲テーブル作成手段と、
前記入力値が入力されたとき、前記範囲テーブルにおいて、入力された前記入力値が含まれる前記範囲を探索し、探索された前記範囲に対応する前記離散値を出力する探索手段と
を備える演算装置。
【請求項２】
前記範囲テーブルに基づいて、ハッシュテーブルを作成するハッシュテーブル作成手段をさらに備え、
前記探索手段は、少なくとも１つの前記範囲を、前記ハッシュテーブルに基づいて、前記範囲テーブルにおける前記範囲の初期探索値を決定し、前記初期探索値と前記範囲テーブルとに基づいて、入力された前記入力値が含まれる前記範囲を探索し、探索された前記範囲に対応する前記離散値を出力する
請求項１に記載の演算装置。
【請求項３】
前記ハッシュテーブル作成手段は、前記範囲テーブルにおいて前記範囲を決めている前記入力値の指数部と仮数部の少なくともいずれか１つを用いて、前記ハッシュテーブルを作成する
請求項２に記載の演算装置。
【請求項４】
前記探索手段は、
出力の頻度の高い離散値に対応する入力値が入力されたとき、前記範囲テーブルに基づいて、入力された前記入力値が含まれる前記範囲を探索し、
出力の頻度の低い離散値に対応する入力値が入力されたとき、前記ハッシュテーブルに基づいて、前記範囲テーブルにおける前記範囲の初期探索値を決定し、前記初期探索値と前記範囲テーブルとに基づいて、入力された前記入力値が含まれる前記範囲を探索する
請求項２に記載の演算装置。
【請求項５】
前記探索手段は、前記入力値が入力されたとき、前記範囲テーブルにおいて、入力された前記入力値が含まれる前記範囲を二分探索する
請求項１に記載の演算装置。
【請求項６】
入力値に対して非線形な演算を施して得られる演算結果を離散化した離散値と、前記離散値をとりうる前記入力値の範囲とを対応付けた範囲テーブルを作成する範囲テーブル作成ステップと、
前記入力値が入力されたとき、前記範囲テーブルにおいて、入力された前記入力値が含まれる前記範囲を探索し、探索された前記範囲に対応する前記離散値を出力する探索ステップと
を含む演算方法。
【請求項７】
入力値に対して非線形な演算を施して得られる演算結果を離散化した離散値と、前記離散値をとりうる前記入力値の範囲とを対応付けた範囲テーブルを作成する範囲テーブル作成ステップと、
前記入力値が入力されたとき、前記範囲テーブルにおいて、入力された前記入力値が含まれる前記範囲を探索し、探索された前記範囲に対応する前記離散値を出力する探索ステップと
を含む処理をコンピュータに実行させるプログラム。

【請求項８】
入力値に対して非線形な演算を施して得られる演算結果を量子化した量子化値と、前記量子化値をとりうる前記入力値の範囲とを対応付けた範囲テーブルを作成する範囲テーブル作成手段と、
前記入力値が入力されたとき、前記範囲テーブルにおいて、入力された前記入力値が含まれる前記範囲を探索し、探索された前記範囲に対応する前記量子化値を出力する探索手段と
を備える量子化装置。
【請求項９】
入力値に対して非線形な演算を施して得られる演算結果を量子化した量子化値と、前記量子化値をとりうる前記入力値の範囲とを対応付けた範囲テーブルを作成する範囲テーブル作成ステップと、
前記入力値が入力されたとき、前記範囲テーブルにおいて、入力された前記入力値が含まれる前記範囲を探索し、探索された前記範囲に対応する前記量子化値を出力する探索ステップと
を含む量子化方法。
【請求項１０】
入力値に対して非線形な演算を施して得られる演算結果を量子化した量子化値と、前記量子化値をとりうる前記入力値の範囲とを対応付けた範囲テーブルを作成する範囲テーブル作成ステップと、
前記入力値が入力されたとき、前記範囲テーブルにおいて、入力された前記入力値が含まれる前記範囲を探索し、探索された前記範囲に対応する前記量子化値を出力する探索ステップと
を含む処理をコンピュータに実行させるプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、演算装置および方法、量子化装置および方法、並びにプログラムに関し、特に、より効率良く非線形な演算を行うことができるようにする演算装置および方法、量子化装置および方法、並びにプログラムに関する。
【背景技術】
【０００２】
アナログ信号をデジタル信号処理で扱うためには、信号の定義域を離散化する標本化処理に加え、信号の値域を離散化する量子化処理が必要である。また、既に値域の離散化が行われたデジタル信号を、より少ないデータ量で扱う場合には、さらに粗く離散化することがあり、この過程も量子化処理に含まれる。量子化処理においては、離散化後の値を量子化値と呼び、その量子化値が割り当てられた符号を量子化符号と呼ぶ。また、隣り合う量子化値の間隔を量子化ステップ幅、値域全体に含まれる量子化値の数を量子化ステップ数とそれぞれ呼ぶ。
【０００３】
一方、デジタル信号処理を行った結果得られるデジタル信号を、アナログ信号として出力する場合等には、量子化符号から、元の連続的な値や小さい量子化ステップ幅で離散化された値を復元する過程が必要となる。この過程を逆量子化処理と呼ぶ。逆量子化処理においては、量子化符号に対応する量子化値を逆量子化変換により変換して元の値に復元する。
【０００４】
ここで、量子化値は、量子化処理において連続的な値を離散的な値に近似したものであるため、逆量子化処理による復元値（逆量子化値）と元の値との間には、量子化処理による誤差が発生する。したがって、逆量子化処理においては、量子化処理が行われる前の値（量子化前の値）を完全に復元することはできず、復元される逆量子化値は、量子化誤差

を含んだ近似値となる。

【０００５】

一般に、量子化ステップ幅が大きい場合、量子化誤差は大きくなるが、量子化ステップ数は少なくなるので、量子化符号を表現するのに必要なデータ量は少なくなる。一方、量子化ステップ幅が小さい場合、量子化誤差は小さくなるが、量子化ステップ数は大きくなるので、量子化符号を表現するのに必要なデータ量は多くなる。

【０００６】

量子化において、量子化ステップ幅が一定であるものを線形量子化と呼ぶのに対して、量子化ステップ幅が一定ではないものを非線形量子化と呼ぶ。非線形量子化は、線形量子化と比較して、下記のような特徴を有している。

【０００７】

まず、信号の出現確率に偏りがある場合、出現の確率が高い値（信号）の近傍では、量子化ステップ幅を小さくし、出現の確率が低い値の近傍では、量子化ステップ幅を大きくした非線形量子化処理を行うことで、線形量子化処理と比較して、平均的に量子化誤差を小さくすることができる。また、同様の手法により、量子化ステップ数を信号の出現確率に応じて調整することにより、線形量子化処理と比較して、平均量子化誤差を悪化させることなく、データ量を削減することもできる。

【０００８】

また、特に、オーディオ信号や画像信号の場合、聴覚や視覚の特性に従い、人間が感覚的に敏感になる値の近傍では、量子化ステップ幅を小さくし、人間が感覚的に敏感でない値の近傍では、量子化ステップ幅を大きくすることで、線形量子化処理と比較して、量子化誤差を人間に知覚させにくくすることができる。また、同様の手法により、量子化ステップ数を人間の知覚の特性に合わせて調整することにより、線形量子化処理と比較して、人間が知覚する誤差を悪化させることなく、データ量を削減することもできる。

【０００９】

したがって、非線形量子化処理は、信号の出現確率に偏りがある信号や、オーディオ信号または画像信号等の人間の感覚的な特性に従う信号等を扱う様々な分野で利用されている。

【００１０】

ところで、オーディオ信号を符号化する方式として、MPEG（Moving Picture Expert Group）オーディオ規格が知られている。MPEGオーディオ規格には複数の符号化方式があるが、ISO/IEC（International Organization for Standardization/International Electrotechnical Commission）13818-7にて、MPEG-2オーディオ規格AAC（Advanced Audio Coding）という符号化方式が標準化されている。

【００１１】

また、さらに拡張されたISO/IEC 14496-3にて、MPEG-4オーディオ規格AACという符号化方式が標準化されている。なお、以下では、MPEG-2オーディオ規格AACとMPEG-4オーディオ規格AACとを併せて、AAC規格と呼ぶこととする。

【００１２】

AAC規格に準拠した従来のオーディオ符号化装置は、聴覚心理モデル保持部、ゲイン制御部、スペクトル処理部、量子化／符号化部、およびマルチプレクサ部を備えている。

【００１３】

オーディオ符号化装置に入力されたオーディオ信号は、聴覚心理モデル保持部において、時間軸に沿ってブロック化され、分割帯域ごとに人間の聴覚特性に従って分析されて、各分割帯域の許容誤差強度が算出される。

【００１４】

また、入力されたオーディオ信号は、ゲイン制御部において、４つの等間隔の周波数帯域に分割され、所定の帯域について利得調整が行われる。

【００１５】

さらに、利得調整されたオーディオ信号は、スペクトル処理部において、周波数領域の

スペクトルデータに変換され、聴覚心理モデル保持部によって算出された許容誤差強度に基づいて、所定の処理が施される。そして、所定の処理が施されたスペクトルデータ（オーディオ信号）は、量子化／符号化部において、符号列に変換され、マルチプレクサ部において、各種の情報が多重化されることで、ビットストリームとして出力される。

【００１６】

上述した量子化／符号化部においては、オーディオ信号に対して非線形量子化処理が行われる。

【００１７】

ところで、上述のようなオーディオ符号化装置によって符号化された符号化ビットストリームに対して、非線形逆量子化処理を行う復号装置が提案されている（例えば、特許文献１参照）。非線形逆量子化処理においては、入力値である量子化値が整数値であるので、入力値としての量子化値と、逆量子化値との関係を予めテーブルとして作成することで、逆量子化処理の実行時には、テーブルに基づいて一意に逆量子化値を決定することができる。

【００１８】

【特許文献１】特開２０００－４７８５０号公報

【発明の開示】

【発明が解決しようとする課題】

【００１９】

しかしながら、非線形量子化処理においては、例えば、入力値が浮動小数点数である場合、非線形逆量子化処理のように、入力値に対して量子化値が一意で決定されるテーブルを作成することはできなかった。

【００２０】

また、上述したオーディオ符号化装置の非線形量子化処理においては、べき乗関数を用いるため、アルゴリズムが難解になり、サイクル数を要してしまう。

【００２１】

上述したようなオーディオ符号化装置に実装される組み込みCPU（Central Processing Unit）やDSP（Digital Signal Processor）は、パーソナルコンピュータのCPUと異なり、動作周波数が数百MHzと低いので、数学ライブラリのようなサイクル数を要する関数を用いることは避けたい。

【００２２】

本発明は、このような状況に鑑みてなされたものであり、より効率良く非線形な演算を行うことができるようにするものである。

【課題を解決するための手段】

【００２３】

本発明の第１の側面の演算装置は、入力値に対して非線形な演算を施して得られる演算結果を離散化した離散値と、前記離散値をとりうる前記入力値の範囲とを対応付けた範囲テーブルを作成する範囲テーブル作成手段と、前記入力値が入力されたとき、前記範囲テーブルにおいて、入力された前記入力値が含まれる前記範囲を探索し、探索された前記範囲に対応する前記離散値を出力する探索手段とを備える。

【００２４】

前記演算装置には、前記範囲テーブルに基づいて、ハッシュテーブルを作成するハッシュテーブル作成手段をさらに設け、前記探索手段には、少なくとも１つの前記範囲を、前記ハッシュテーブルに基づいて、前記範囲テーブルにおける前記範囲の初期探索値を決定させ、前記初期探索値と前記範囲テーブルとに基づいて、入力された前記入力値が含まれる前記範囲を探索させ、探索された前記範囲に対応する前記離散値を出力させることができる。

【００２５】

前記ハッシュテーブル作成手段には、前記範囲テーブルにおいて前記範囲を決めている前記入力値の指数部と仮数部の少なくともいずれか１つを用いて、前記ハッシュテーブル

を作成させることができる。

【００２６】

前記探索手段には、出力の頻度の高い離散値に対応する入力値が入力されたとき、前記範囲テーブルに基づいて、入力された前記入力値が含まれる前記範囲を探索させ、出力の頻度の低い離散値に対応する入力値が入力されたとき、前記ハッシュテーブルに基づいて、前記範囲テーブルにおける前記範囲の初期探索値を決定し、前記初期探索値と前記範囲テーブルとに基づいて、入力された前記入力値が含まれる前記範囲を探索させることができる。

【００２７】

前記探索手段には、前記入力値が入力されたとき、前記範囲テーブルにおいて、入力された前記入力値が含まれる前記範囲を二分探索させることができる。

【００２８】

本発明の第１の側面の演算方法は、入力値に対して非線形な演算を施して得られる演算結果を離散化した離散値と、前記離散値をとりうる前記入力値の範囲とを対応付けた範囲テーブルを作成する範囲テーブル作成ステップと、前記入力値が入力されたとき、前記範囲テーブルにおいて、入力された前記入力値が含まれる前記範囲を探索し、探索された前記範囲に対応する前記離散値を出力する探索ステップとを含む。

【００２９】

本発明の第１の側面のプログラムは、入力値に対して非線形な演算を施して得られる演算結果を離散化した離散値と、前記離散値をとりうる前記入力値の範囲とを対応付けた範囲テーブルを作成する範囲テーブル作成ステップと、前記入力値が入力されたとき、前記範囲テーブルにおいて、入力された前記入力値が含まれる前記範囲を探索し、探索された前記範囲に対応する前記離散値を出力する探索ステップとを含む処理をコンピュータに実行させる。

【００３０】

本発明の第２の側面の量子化装置は、入力値に対して非線形な演算を施して得られる演算結果を量子化した量子化値と、前記量子化値をとりうる前記入力値の範囲とを対応付けた範囲テーブルを作成する範囲テーブル作成手段と、前記入力値が入力されたとき、前記範囲テーブルにおいて、入力された前記入力値が含まれる前記範囲を探索し、探索された前記範囲に対応する前記量子化値を出力する探索手段とを備える。

【００３１】

本発明の第２の側面の量子化方法は、入力値に対して非線形な演算を施して得られる演算結果を量子化した量子化値と、前記量子化値をとりうる前記入力値の範囲とを対応付けた範囲テーブルを作成する範囲テーブル作成ステップと、前記入力値が入力されたとき、前記範囲テーブルにおいて、入力された前記入力値が含まれる前記範囲を探索し、探索された前記範囲に対応する前記量子化値を出力する探索ステップとを含む。

【００３２】

本発明の第２の側面のプログラムは、入力値に対して非線形な演算を施して得られる演算結果を量子化した量子化値と、前記量子化値をとりうる前記入力値の範囲とを対応付けた範囲テーブルを作成する範囲テーブル作成ステップと、前記入力値が入力されたとき、前記範囲テーブルにおいて、入力された前記入力値が含まれる前記範囲を探索し、探索された前記範囲に対応する前記量子化値を出力する探索ステップとを含む処理をコンピュータに実行させる。

【００３３】

本発明の第１の側面においては、入力値に対して非線形な演算を施して得られる演算結果を離散化した離散値と、離散値をとりうる入力値の範囲とを対応付けた範囲テーブルが作成され、入力値が入力されたとき、範囲テーブルにおいて、入力された入力値が含まれる範囲が探索され、探索された範囲に対応する離散値が出力される。

【００３４】

本発明の第２の側面においては、入力値に対して非線形な演算を施して得られる演算結

果を量子化した量子化値と、量子化値をとりうる入力値の範囲とを対応付けた範囲テーブルが作成され、入力値が入力されたとき、範囲テーブルにおいて、入力された入力値が含まれる範囲が探索され、探索された範囲に対応する量子化値が出力される。

【発明の効果】

【００３５】

本発明の第１および第２の側面によれば、より効率良く非線形な演算を行うことが可能となる。

【発明を実施するための最良の形態】

【００３６】

以下、本発明の実施の形態について図を参照して説明する。なお、説明は以下の順序で行う。

１．第１の実施の形態

２．第２の実施の形態

３．第３の実施の形態

【００３７】

＜１．第１の実施の形態＞

［オーディオ符号化装置の構成例］

図１に、本発明を適用したオーディオ符号化装置の一実施の形態の構成例を示す。

【００３８】

図１のオーディオ符号化装置は、聴覚心理モデル保持部１１、ゲイン制御部１２、スペクトル処理部１３、量子化／符号化部１４、およびマルチプレクサ部１５から構成される。

【００３９】

オーディオ符号化装置に入力されたオーディオ信号は、聴覚心理モデル保持部１１およびゲイン制御部１２に供給される。聴覚心理モデル保持部１１は、入力されたオーディオ信号を時間軸に沿ってブロック化し、ブロック化されたオーディオ信号を分割帯域ごとに人間の聴覚特性に従って分析して、各分割帯域の許容誤差強度を算出する。聴覚心理モデル保持部１１は、算出した許容誤差強度を、スペクトル処理部１３、および、量子化／符号化部１４に供給する。

【００４０】

ゲイン制御部１２は、AAC規格の符号化アルゴリズムとして用意されているMain, LC（Low Complexity），SSR（Scalable Sampling Rate）の３つのプロファイルのうちの、SSRプロファイルのみに使用される。ゲイン制御部１２は、入力されたオーディオ信号を４つの等間隔の周波数帯域に分割し、例えば、最低域以外の帯域について利得調整を行って、スペクトル処理部１３に供給する。

【００４１】

スペクトル処理部１３は、ゲイン制御部１２において利得調整されたオーディオ信号を周波数領域のスペクトルデータに変換する。また、スペクトル処理部１３は、聴覚心理モデル保持部１１から供給された許容誤差強度に基づいて、スペクトル処理部１３の各部を制御し、スペクトルデータに所定の処理を施す。

【００４２】

スペクトル処理部１３は、MDCT（Modified Discrete Cosine Transform）部２１，TNS（Temporal Noise Shaping）処理部２２、インテンシティ／カップリング部２３、予測部２４、およびM/Sステレオ（Middle/Side Stereo）部２５を備えている。

【００４３】

MDCT部２１は、ゲイン制御部１２から供給された時間領域のオーディオ信号を周波数領域のスペクトルデータ（MDCT係数）に変換し、TNS処理部２２に供給する。TNS処理部２２は、MDCT部２１からのスペクトルデータを、時間領域の信号であるかのように見立てて線形予測を行い、スペクトルデータに対して予測フィルタリングを行って、ビットストリームとして、インテンシティ／カップリング部２３に供給する。インテンシティ／カップリ

ング部２３は、TNS処理部２２からのスペクトルデータとしてのオーディオ信号に対して、異なるチャンネルの相関を利用した圧縮処理（ステレオ相関符号化処理）を施す。

【００４４】

予測部２４は、上述した３つのプロファイルのうちの、Mainプロファイルのみに使用される。予測部２４は、インテンシティ／カップリング部２３においてステレオ相関符号化されたオーディオ信号と、量子化／符号化部１４から供給されるオーディオ信号とを用いて予測符号化を行い、その結果得られたオーディオ信号をM/Sステレオ部２５に供給する。M/Sステレオ部２５は、予測部２４からのオーディオ信号をステレオ相関符号化し、量子化／符号化部１４に供給する。

【００４５】

量子化／符号化部１４は、正規化係数部３１、量子化部３２、およびハフマン符号化部３３を備えており、スペクトル処理部１３のM/Sステレオ部２５からのオーディオ信号を符号列に変換し、マルチプレクサ部１５に供給する

【００４６】

正規化係数部３１は、M/Sステレオ部２５からのオーディオ信号を量子化部３２に供給するとともに、そのオーディオ信号に基づいて、オーディオ信号の量子化に用いる正規化係数を算出し、量子化部３２およびハフマン符号化部３３に供給する。図１の量子化装置においては、例えば、聴覚心理モデル保持部１１からの許容誤差強度が用いられて、分割帯域ごとの正規化係数として、量子化ステップパラメータが算出される。

【００４７】

量子化部３２は、正規化係数部３１からの正規化係数を用いて、正規化係数部３１から供給されたオーディオ信号を非線形量子化し、その結果得られたオーディオ信号（量子化値）をハフマン符号化部３３および予測部２４に供給する。ハフマン符号化部３３は、予め定められたハフマン符号表に基づいて、正規化係数部３１からの正規化係数と、量子化部３２からの量子化値とをハフマン符号化し、マルチプレクサ部１５に供給する。

【００４８】

マルチプレクサ部１５は、ゲイン制御部１２およびMDCT部２１乃至正規化係数部３１からそれぞれ供給された、オーディオ信号の符号化の過程で生成された各種の情報と、ハフマン符号化部３３からのハフマン符号とを多重化する。これにより、マルチプレクサ部１５は、オーディオ信号のビットストリームを生成し、生成したオーディオ信号のビットストリームを出力する。

【００４９】

上述した量子化部３２におけるオーディオ信号の非線形量子化処理は、正規化係数部３１からのオーディオ信号の値を入力値Ｘとし、正規化係数部３１からの量子化ステップ幅を表す量子化ステップパラメータをｑとすると、式（１）を計算することにより行われる。すなわち、式（１）が計算されて、オーディオ信号としての入力値Ｘから量子化値Ｚが求められる。

【００５０】

【数１】

$$Z = (\mathrm{int})\left(\left(\frac{X}{2^{q/4}}\right)^{3/4} - 0.0946 + 0.5\right) \qquad \cdots (1)$$

【００５１】

なお、式（１）において、(int)(Ａ)は、浮動小数点数Ａの小数部分を切り捨てて整数部分を得る演算を示す。

【００５２】

AAC規格の符号化においては、量子化値Ｚは１４ビットと規定されているため、量子化前の値Ｙ＝Ｘ／$2^{q/4}$とすると、量子化前の値Ｙは、逆算することで、０≦Ｙ＜$8191.5943^{(4/3)}$の範囲の値とされる。したがって、量子化値Ｚは、０≦Ｚ≦8191の範囲の整数とな

り、量子化値Ｚがこの範囲内の値となるように、量子化ステップパラメータｑが定められる。

【００５３】

また、オーディオ信号の量子化処理において、量子化部３２は、全ての入力値Ｘに対する量子化値Ｚを逆量子化し、量子化誤差が所定の範囲内に収まっているか否かを確認する。例えば、式（２）の演算により逆量子化が行われ、逆量子化値Ｗが得られる。

【００５４】

【数２】

$$W = (Z)^{4/3} \qquad \cdots (2)$$

【００５５】

さらに、量子化誤差は、逆量子化値Ｗと入力値Ｘとの差分を求めることで得られ、この差分が所定の範囲内の値となっているか否かの判定が行われる。

【００５６】

すなわち、非線形逆量子化処理においては、入力値である量子化値Ｚが整数値であるので、量子化値Ｚと逆量子化値Ｗとの関係を予めテーブルとして作成することで、逆量子化処理の実行時には、テーブルに基づいて一意に逆量子化値を決定することができる。一方、非線形量子化処理においては、式（１）における入力値Ｘが浮動小数点数である場合、量子化前の値Ｙ＝Ｘ／$2^{q/4}$も浮動小数点数となるので、量子化前の値Ｙに対して量子化値Ｚが一意で決定されるテーブルを作成することはできなかった。

【００５７】

［量子化部の構成例］

そこで、図２のブロック図を参照して、量子化処理において、より効率良く非線形な演算を行うようにした量子化部３２の構成例について説明する。

【００５８】

図２の量子化部３２は、範囲テーブル作成部７１および探索量子化部７２から構成され、正規化係数部３１からのオーディオ信号としての入力データＸに基づいて、量子化値Ｚを出力する。

【００５９】

範囲テーブル作成部７１は、以下に示される式（３）において、量子化前の値Ｙを順次変化（増加）させていき、量子化値Ｚが変化したときの量子化前の値Ｙと、その時の量子化値Ｚとを対応付けた範囲テーブルを作成し、探索量子化部７２に供給する。より具体的には、範囲テーブル作成部７１は、量子化前の値Ｙを、0.00000000から0.00000001，0.00000002，0.00000003，・・・と、順次増加させていき、離散値としての量子化値Ｚが変化したときの量子化前の値Ｙを求める。そして、範囲テーブル作成部７１は、量子化値Ｚが変化したときの量子化前の値Ｙと、その時の量子化値Ｚとを対応付けた範囲テーブルを作成する。

【００６０】

【数３】

$$Z = (\mathrm{int})\left(Y^{3/4} - 0.0946 + 0.5\right) \qquad \cdots (3)$$

【００６１】

量子化値Z[m]に対応する量子化前の値をY[m]とすると、量子化値Z[m]をとりうる量子化前の値Ｙの範囲は、Y[m-1]≦Ｙ＜Y[m]となる（ｍは、入力データＸの識別子であり、０以上の整数）。すなわち、例えば、範囲テーブル作成部７１は、量子化値Z[m]と、量子化前の値Ｙの範囲Y[m-1]≦Ｙ＜Y[m]とを対応付けた範囲テーブルを作成する。なお、以下では、量子化値Z[m]と、量子化前の値Ｙの範囲Y[m-1]≦Ｙ＜Y[m]とが対応付けられた範囲テーブルを、単に、量子化値Z[m]と、量子化前の値Ｙの範囲（の境界値）とが対応付けられた範囲テーブルということとする。

【００６２】

探索量子化部７２は、探索部７２ａおよび決定部７２ｂを備えている。探索部７２ａは、範囲テーブル作成部７１から供給された範囲テーブルにおいて、正規化係数部３１から供給された入力データＸから算出される量子化前の値Ｙが含まれる範囲を探索する。決定部７２ｂは、探索部７２ａによって探索された範囲に対応付けられている量子化値Ｚを決定し、ハフマン符号化部３３に供給する。

【００６３】

以上の構成により、量子化部３２は、予め作成した範囲テーブルに基づいて、入力された入力データから得られる量子化前の値に対応する範囲を探索し、量子化値を出力する。

【００６４】

［量子化部の範囲テーブル作成処理］

次に、図３のフローチャートを参照して、図２の量子化部３２の範囲テーブル作成処理について説明する。量子化部３２において、範囲テーブル作成処理は、量子化処理を行う前に行われる。

【００６５】

ステップＳ５１において、範囲テーブル作成部７１は、上述した式（３）において、量子化前の値Ｙを順次増加させていき、量子化値Ｚが変化したときの量子化前の値Ｙと、その時の量子化値Ｚとを対応付けた範囲テーブルを作成し、探索量子化部７２に供給する。

【００６６】

例えば、範囲テーブル作成部７１は、図４に示されるように、量子化値Ｚと、量子化前の値Ｙの範囲（の境界値）とを対応付けた範囲テーブルを作成する。量子化値Ｚは、０≦Ｚ≦8191の範囲の整数であり、図４においては、それぞれの量子化値Ｚについて、式（３）を基に得られた量子化前の値Ｙの範囲（の境界値）が対応付けられている。

【００６７】

以上の処理により、量子化処理を行う前に、量子化前の値Ｙの範囲と、量子化値Ｚとを対応付けた範囲テーブルを作成することができる。

【００６８】

［量子化部の量子化処理］

次に、図５のフローチャートを参照して、図２の量子化部３２の量子化処理について説明する。

【００６９】

ステップＳ７１において、探索量子化部７２の探索部７２ａは、範囲テーブル作成部７１から予め供給された範囲テーブルにおいて、正規化係数部３１から供給された入力データＸから算出される量子化前の値Ｙが含まれる範囲を探索する。より具体的には、探索量子化部７２は、正規化係数部３１からの入力データＸから、量子化前の値$Y = X / 2^{q/4}$を算出し、算出した量子化前の値Ｙが含まれる範囲を、範囲テーブルにおいて探索する。

【００７０】

ステップＳ７２において、探索量子化部７２の決定部７２ｂは、探索部７２ａによって決定された範囲に対応付けられている量子化値Ｚを決定し、ハフマン符号化部３３に供給する。

【００７１】

図６は、図５のフローチャートのステップＳ７１，Ｓ７２における処理をＣ言語で記述したプログラムの例を示している。

【００７２】

図６のプログラム８１において、各行の左端の数字は、各行の行番号を示しており、説明のために設けたものである。つまり、実際の記述においては不要である。以降、他のプログラムの例においても同様であるものとする。

【００７３】

プログラム８１の１行目は、入力データの識別子を表すｍを０からdata_size個まで１ずつインクリメントし、以下の処理を繰り返すことを定義している。

【００７４】

プログラム８１の２，３行目においては、ｍ番目に入力された量子化前の値Y[m]が、4.9999600e-01Fより小さいか否かが判定される。量子化前の値Y[m]が、4.9999600e-01Fより小さい場合、範囲テーブルにおいて、量子化前の値の範囲Y[m]＜4.9999600e-01Fに対応付けられている量子化値Z[m]=0が決定される。

【００７５】

一方、量子化前の値Y[m]が、4.9999600e-01Fより小さくない場合、５，６行目においては、ｍ番目に入力された量子化前の値Y[m]が、1.8629548e+00Fより小さいか否かが判定される。量子化前の値Y[m]が、1.8629548e+00Fより小さい場合、範囲テーブルにおいて、量子化前の値の範囲Y[m]＜1.8629548e+00Fに対応付けられている量子化値Z[m]=1が決定される。

【００７６】

以降、同様にして、N-2行目まで、量子化前の値Y[m]が含まれる範囲が、範囲テーブルにおいて、小さな値から順次探索され（ステップＳ７１の処理）、その量子化前の値Y[m]に対応した範囲に設定されている量子化値Z[m]が決定される（ステップＳ７２の処理）。

【００７７】

このようにして、非線形な演算をすることなく、最大でも8192回の探索によって、量子化値を決定することができる。

【００７８】

図５のフローチャートに戻り、ステップＳ７３において、探索量子化部７２は、data_size個全ての入力データに対して、量子化値Ｚの探索の処理を終えたか否かを判定する。全ての入力データに対して、量子化値Ｚの探索の処理を終えていないと判定された場合、処理は、ステップＳ７１に戻り、data_size個全ての入力データに対して処理を終えるまで、ステップＳ７１乃至Ｓ７３を繰り返す。

【００７９】

以上の処理により、予め求められている、量子化値Ｚと、量子化値Ｚをとりうる量子化前の値Ｙの範囲とが対応付けられた範囲テーブルに基づいて、入力された入力データＸから算出される量子化前の値Ｙに対応する量子化値Ｚを決定できる。したがって、式（３）に含まれる、非線形関数である３／４乗のべき乗関数の演算をすることなく、図６で説明したような、条件による探索によって量子化値Ｚを決定できるので、より効率良く非線形な演算を行うことが可能となる。

【００８０】

以上においては、8192個の条件を順番に探索することによって量子化値を決定する例について説明してきたが、例えば、二分探索を用いれば、2^{13}＝8192であるので、１３回の条件の判定で、量子化値を決定することができる。

【００８１】

ところで、図７の量子化値Ｚに対するヒストグラムに示されるように、量子化値Ｚについて、Ｚ＜１０程度の量子化値が得られる量子化処理の頻度は高く、量子化値Ｚ＝０となる量子化処理の頻度が最も高い。また、量子化値Ｚが大きくなるにつれて、その量子化値Ｚが得られる量子化処理の頻度は低くなっている。すなわち、図７は、量子化値Ｚが小さい程、その量子化値Ｚが得られる量子化処理の回数が多いことを示している。なお、図７においては、横軸が量子化値Ｚを示しており、縦軸が量子化処理の頻度（回数）を示している

【００８２】

以下では、量子化処理の頻度の傾向を利用した量子化部の例について説明する。

【００８３】

＜２．第２の実施の形態＞

［量子化部の構成例］

図８は、量子化処理の頻度の傾向を利用した量子化部の構成例を示している。なお、図８の量子化部１５１において、図２の量子化部３２に設けられたものと同様の機能を備える構成については、同一名称および同一符号を付するものとし、その説明は、適宜省略す

るものとする。

【００８４】

すなわち、図８の量子化部１５１において、図２の量子化部３２と異なるのは、ハッシュテーブル作成部１７１を新たに設け、探索量子化部７２に代えて、探索量子化部１７２を設けた点である。

【００８５】

図８の量子化部１５１においては、範囲テーブル作成部７１は、作成した範囲テーブルを、ハッシュテーブル作成部１７１および探索量子化部１７２に供給する。

【００８６】

ハッシュテーブル作成部１７１は、範囲テーブル作成部７１からの範囲テーブルに基づいて、テーブル値の探索を高速にするハッシュテーブルを作成し、探索量子化部１７２に供給する。

【００８７】

ここで、ハッシュテーブルとは、範囲テーブルのテーブル値である量子化前の値が含まれる範囲を、量子化前の値に応じたグループに分け、そのグループを示す情報をテーブル値とするテーブルである。すなわち、ハッシュテーブルによれば、量子化前の値が入力されたとき、その量子化前の値に対応したグループが決定され、そのグループにおいて、最初に探索すべきテーブル値である初期探索値から探索が開始される。したがって、範囲テーブルに定義された全てのテーブル値を順次探索するより高速にテーブル値を探索することができる。なお、ハッシュテーブルの作成の詳細は後述する。

【００８８】

探索量子化部１７２は、初期探索値決定部１７２ａ、探索部１７２ｂ、および決定部１７２ｃを備えている。初期探索値決定部１７２ａは、ハッシュテーブル作成部１７１から供給されたハッシュテーブルを用いて、範囲テーブルにおける探索開始の値（初期探索値）を決定する。探索部１７２ｂは、範囲テーブル作成部７１から供給された範囲テーブルにおいて、正規化係数部３１から供給された入力データＸから算出された量子化前の値Ｙが含まれる範囲を、初期探索値決定部１７２ａにより決定された初期探索値から探索する。決定部１７２ｃは、探索部１７２ｂによって探索された範囲に対応付けられている量子化値Ｚを決定し、ハフマン符号化部３３に供給する。

【００８９】

以上の構成により、量子化部１５１は、予め作成した範囲テーブルとハッシュテーブルとに基づいて、入力された入力データから得られる量子化前の値に対応する量子化値を探索し、決定する。

【００９０】

［量子化部のハッシュテーブル作成処理］

次に、図９のフローチャートを参照して、図８の量子化部１５１のハッシュテーブル作成処理について説明する。量子化部１５１において、ハッシュテーブル作成処理は、量子化処理を行う前に行われる。なお、図９のフローチャートにおけるステップＳ１５１の処理は、図４のフローチャートを参照して説明した範囲テーブル作成処理と同様であるので、その説明は省略するものとする。

【００９１】

すなわち、ステップＳ１５２において、ハッシュテーブル作成部１７１は、範囲テーブル作成部７１からの範囲テーブルに基づいて、ハッシュテーブルを作成し、探索量子化部１７２に供給する。

【００９２】

より具体的には、例えば、ハッシュテーブル作成部１７１は、IEEE（Institute for Electrical and Electronics Engineering）754に基づく浮動小数点数の指数部と仮数部を用いてハッシュテーブルを作成する。

【００９３】

IEEE754の規格に基づく単精度浮動小数点数を示す浮動小数点型のデータは、図１０に

示されるように、３２ビットのビット列で構成される。すなわち、浮動小数点数のデータの最下位ビットである０ビット目から２２ビット目までの部分は、仮数部の仮数Ｆとされ、２３ビット目から３０ビット目までは指数部の指数Ｅとされ、最上位の３１ビット目は符号ビットＳとされる。

【００９４】

ここで、符号ビットＳは、仮数部が正である場合には「０」とされ、仮数部が負の場合には「１」とされる。また、指数部のビット構成は「指数＋バイアス」となる。例えば、IEEE754の規格では、単精度のバイアスとして１２７が用いられているため、指数の数値が「０」であるときは、指数部の値は「１２７」（＝０＋１２７）となり、指数部は１２７のビット構成（０ｘ７ｆ）となる（「０ｘ」は「７ｆ」が１６進数であることを示す）。

【００９５】

なお、指数部が０と２５５の場合は、特別な値の意味を持つが、その説明は省略する。

【００９６】

このような浮動小数点型のデータにより表される値は、数値表現形式で表すと、式（４）に示す通りとなる。

【００９７】

【数４】

$$(-1)^{S}\times 2^{(E-B)}\times(1+.F) \qquad \cdots (4)$$

【００９８】

なお、式（４）において、「．Ｆ」は、仮数Ｆの上位ビットの前に小数点が位置するものとして仮数部を小数点表現していることを示す。また、式（４）における「Ｂ」はバイアス成分を示しており、「Ｓ」は符号ビットを示している。

【００９９】

次に、上述した浮動小数点数の指数部および仮数部を用いたハッシュテーブルの作成手順の一例を説明する。

【０１００】

ハッシュテーブル作成部１７１は、指数部について、８ビットの指数部の値から125を減算し、その値をindex1とする。ここでは、量子化値Ｚ＝０となる範囲が０≦Ｙ＜0.4999600であり、0.25≦0.4999600＜0.5を満たし、0.25乃至0.5の区間は指数部の値が125となるので、指数部の減算値を125とする。また、量子化値Ｚ＝8191となる範囲は、$1.6512946\times10^{5}\leqq$Ｙであり、$1.6512946\times10^{5}$を単精度浮動小数点数で表現したときの指数部は144となる。index1は、０乃至１９の値をとる。

【０１０１】

また、ハッシュテーブル作成部１７１は、仮数部について、２３ビットの仮数部のうちの上位４ビットを、index2とする。index2は、０乃至１５の値をとる。

【０１０２】

index1およびindex2を要素数とする２次元配列としてのハッシュテーブルに対して、ハッシュテーブル作成部１７１は、8192個の範囲テーブルの値から指数部および仮数部を取り出し、index1とindex2のいずれかの値が変化する範囲テーブルのインデックスを要素として、ハッシュテーブル（２次元配列）に配置していく。今回の場合、量子化値が０から始まっているので、範囲テーブルのインデックスと量子化値Ｚは同じものになる。

【０１０３】

後述する図１３における範囲テーブルquantize_tab[8192]を用いて、ハッシュテーブルの作成の手順について説明する。インデックス１のquantize_tab[1]=1.8629548は、index1=3，index2=12となり、インデックス２のquantize_tab[2]=3.5652816は、index1=4，index2=6となる。これにより量子化値Ｚ＝２となる範囲は1.8629548≦Ｙ＜3.5652816であるので、この区間の量子化前の値Ｙに対するindex1とindex2の組み合わせはindex1=3，index2=12乃至15、またはindex1=4，index2=0乃至6のいずれかになる。これらのindex1とinde

x2のハッシュテーブルには範囲テーブルのインデックス２を配置しておく。

【０１０４】

図１１は、後述する図１３における範囲テーブルquantize_tab[8192]に基づいて、上述した手順で作成されるハッシュテーブルquantize_hash[index1][index2]をＣ言語で記述した例を示している。

【０１０５】

図１１のプログラム１８１において、quantize_hash[20][16]は、２０×１６の２次元配列を表している。

【０１０６】

例えば、３，４行目には、index1=０，index2=０乃至１６であるテーブル値が記述されている。６，７行目には、index1=１，index2=０乃至１６であるテーブル値が記述されている。以降、同様にして、テーブル値が記述され、N-3，N-2行目には、index1=１９，index2=０乃至１６であるテーブル値が記述されている。

【０１０７】

図１１のプログラム１８１に示されるとおり、ハッシュテーブルに指数部および仮数部を用いると、ハッシュテーブルのテーブル値（要素）として、量子化値が小さいものが多く割り当てられている。

【０１０８】

特に、３乃至２８行目のindex1 = 8以下の場合、隣り合う値との差が１しかないので、後述する量子化処理において、index1 = 8以下となる量子化前の値Ｙについては、１回のみの探索で量子化値Ｚを決定することができる。

【０１０９】

また、上述のハッシュテーブルを用いた場合、例えば、index1 = 18，index2 = 0のとき、そのときのテーブル値4096と、次のテーブル値4286との差（量子化前の値の範囲）が190あるので、190回の探索を行う必要がある。しかしながら、図７で示したように、このテーブル値付近の量子化値はほとんど発生しないので、探索の回数が多くなったとしても問題ではない。

【０１１０】

以上の処理により、量子化処理を行う前に、範囲テーブルにおけるテーブル値の探索を高速にするハッシュテーブルを作成することができる。

【０１１１】

［量子化部の量子化処理］

次に、図１２のフローチャートを参照して、図８の量子化部１５１の量子化処理について説明する。

【０１１２】

ステップＳ１７１において、探索量子化部１７２の探索部１７２ｂは、範囲テーブル作成部７１から予め供給された範囲テーブルに基づいて、正規化係数部３１から供給された入力データＸに対応する量子化前の値Ｙが含まれる範囲を、条件文を用いて所定回数だけ探索する。より具体的には、探索部１７２ｂは、正規化係数部３１からの入力データＸから、量子化前の値$Y = X / 2^{q/4}$を算出し、算出した量子化前の値Ｙが含まれる範囲を、範囲テーブルにおいて、条件文を用いて所定回数だけ探索する。

【０１１３】

ステップＳ１７２において、探索部１７２ｂは、所定回数の探索によって、量子化前の値Ｙが含まれる範囲が求まったか否かを判定する。

【０１１４】

ステップＳ１７２において、所定回数の探索によって、量子化前の値Ｙが含まれる範囲が求まったと判定された場合、処理は、ステップＳ１７３に進む。

【０１１５】

ステップＳ１７３において、探索量子化部１７２の決定部１７２ｃは、探索部１７２ｂによって探索された範囲に対応付けられている量子化値を決定し、ハフマン符号化部３３

に供給して、処理は、ステップＳ１７７に進む。

【０１１６】

一方、ステップＳ１７２において、所定回数の探索によって、量子化前の値Ｙが含まれる範囲が求まらなかった場合、処理は、ステップＳ１７４に進む。

【０１１７】

ステップＳ１７４において、探索量子化部１７２の初期探索値決定部１７２ａは、ハッシュテーブル作成部１７１から供給されたハッシュテーブルを用いて、範囲テーブルにおける初期探索値を決定する。より具体的には、初期探索値決定部１７２ａは、正規化係数部３１からの入力データＸから、量子化前の値$Y = X / 2^{q/4}$を算出し、算出した量子化前の値Ｙ（浮動小数点数）の指数部および仮数部に基づいて、index1とindex2を算出し、ハッシュテーブルから範囲テーブルの初期探索値を決定する。

【０１１８】

ステップＳ１７５において、探索部１７２ｂは、初期探索値と、範囲テーブル作成部７１から供給された範囲テーブルとに基づいて、入力データＸに対応する量子化前の値Ｙが含まれる範囲を探索する。

【０１１９】

ステップＳ１７６において、決定部１７２ｃは、探索部１７２ｂによって探索された範囲に対応付けられている量子化値Ｚを決定し、ハフマン符号化部３３に供給する。

【０１２０】

ステップＳ１７７において、探索量子化部１７２は、data_size個全ての入力データに対して、量子化値Ｚの探索の処理を終えたか否かを判定する。全ての入力データに対して、量子化値Ｚの探索の処理を終えていないと判定された場合、処理は、ステップＳ１７１に戻り、data_size個全ての入力データに対して処理を終えるまで、ステップＳ１７１乃至Ｓ１７７を繰り返す。

【０１２１】

図１３は、図１２のフローチャートのステップＳ１７１乃至Ｓ１７７における処理をＣ言語で記述したプログラムの例を示している。

【０１２２】

図１３のプログラム１９１において、１行目乃至8195行目のquantize_tab[8192]は、範囲テーブルを示している。

【０１２３】

8196行目乃至8199行目は、値uni0が異なる型のデータとして用いられることを定義している。また、8200行目は、入力データの識別子を表すｍを０からdata_size個まで１ずつインクリメントし、以下の処理を繰り返すことを定義している。

【０１２４】

8202行目乃至8215行目においては、量子化前の値Y[m]（=uni0.f）が含まれる範囲が順番に５回、条件文により探索される（ステップＳ１７１の処理）。

【０１２５】

また、8218行目乃至8220行目においては、図１１で説明したハッシュテーブルquantize_hash[index1][index2]を用いて、量子化前の値Y[m]（=uni0.f）の初期探索値が決定される（ステップＳ１７４の処理）。

【０１２６】

そして、8221行目および8222行目においては、初期探索値ｋと範囲テーブルquantize_tab[k]とに基づいて、量子化前の値Y[m]（=uni0.f）が含まれる範囲が探索される（ステップＳ１７５の処理）。

【０１２７】

このようにして、量子化前の値が大きい場合には、ハッシュテーブルによって初期探索値が決定されるので、小さな値から順次探索していく図６のプログラム８１よりも高速に、量子化前の値が含まれる範囲が探索される。

【０１２８】

以上の処理により、量子化処理の頻度の高い量子化値に対しては範囲テーブルに基づいて、また、量子化処理の頻度の低い量子化値に対してはハッシュテーブルに基づいて、入力された入力データＸから算出される量子化前の値Ｙに対応する量子化値Ｚを決定できる。したがって、式（３）で示される、非線形関数である３／４乗のべき乗関数の演算をすることなく、また、より少ない回数の探索によって量子化値Ｚを決定できるので、より効率良く、かつ、より迅速に非線形な演算を行うことが可能となる。

【０１２９】

なお、以上においては、範囲テーブルに基づく条件文判定と、ハッシュテーブルによる初期探索値の決定とを併用して量子化値Ｚを求めるようにしたが、ハッシュテーブルによる初期探索値の決定のみで量子化値Ｚを求めるようにしてもよい。

【０１３０】

［固定小数点数での応用例］

以上においては、量子化前の値や範囲テーブルの値を、浮動小数点数として扱ってきたが、これらの値を固定小数点数として扱うこともできる。より具体的には、量子化値に対応する量子化前の値の範囲を浮動小数点数で算出し、その浮動小数点数を基に、固定小数点数の整数部分を算出すればよい。

【０１３１】

以下に、上述した、浮動小数点数で表現された範囲テーブルquantize_tab[8192]に対して、３２ビットのうちの８ビット目に小数点の位置を設けて、固定小数点数とした場合の、範囲テーブルとハッシュテーブルを併用した例について説明する。範囲テーブル値は、小数点の位置が８ビット目なので、256倍して整数部分のみをとることで作成される。

【０１３２】

まず、初期探索値決定部１７２ａにおけるハッシュテーブルのindex1およびindex2の算出は、以下の手順で行われる。

【０１３３】

すなわち、量子化前の値Ｙにおいて、１がたつ最上位ビットの位置cntを探索し、index1は、式（５）より算出される。

【０１３４】

【数５】

$$index1 = cnt - P + 2 \qquad \cdots (5)$$

【０１３５】

式（５）において、Ｐは、小数点位置を示しており、この場合、Ｐ＝８である。また、式（５）における“２”は、浮動小数点数のハッシュテーブル作成の際に、指数部の基準となった125と、IEEE754の規格で定められているバイアス127との差分である。

【０１３６】

また、index2は、式（６）および式（７）で示されるように、量子化前の値Ｙにおいて、１がたつ最上位ビットを０にして、cnt－４分右シフトすることで算出されるものとする。

【０１３７】

【数６】

$$Y = Y - mask \qquad \cdots (6)$$

$$index2 = Y >> (cnt - 4) \qquad \cdots (7)$$

【０１３８】

なお、式（６）における“mask”は、１６進数で「0x40000000」（２進数で「100 0000 0000 0000 0000 0000 0000 0000」（３２ビット））から始まり、量子化前の値Ｙに対して１がたつ最上位ビットまで１ビットずつ右シフト演算されていく。“mask”は、最終的に、量子化前の値Ｙに対して１がたつ最上位ビットを表す値になる。すなわち、式（６）で減算することで、１がたつ最上位ビットを０にしている。式（７）における“>>”は、

右シフト演算を表している。

【０１３９】

図１４は、図１３で説明した範囲テーブルquantize_tab[8192]における浮動小数点数の値を３２ビットで８ビット目に小数点の位置がある固定小数点数とした場合の例をＣ言語で記述したプログラムの例を示している。

【０１４０】

図１４のプログラム２０１において、１行目乃至8195行目のquantize_tab_int[8192]は、固定小数点数化された範囲テーブルを示している。

【０１４１】

8196行目は、整数型の値yinを定義している。また、8197行目は、入力データの識別子を表すｍを０からdata_size個まで１ずつインクリメントし、以下の処理を繰り返すことを定義している。

【０１４２】

8199行目乃至8212行目においては、３２ビットで８ビット目に小数点の位置がある、固定小数点数化された量子化前の値Y[m]（=yin）が含まれる範囲が順番に５回探索される（ステップＳ１７１の処理）。

【０１４３】

また、8223行目乃至8225行目においては、図１１で説明したハッシュテーブルquantize_hash[index1][index2]を用いて、量子化前の値Y[m]（=yin）の初期探索値が決定される（ステップＳ１７４の処理）。

【０１４４】

そして、8226行目および8227行目においては、初期探索値ｋと範囲テーブルquantize_tab[k]とに基づいて、量子化前の値Y[m]（=yin）が含まれる範囲が探索される（ステップＳ１７５の処理）。

【０１４５】

以上の処理においても、式（１）に含まれるべき乗関数の演算をすることなく、固定小数点数の値からなる範囲テーブルおよびハッシュテーブルを用いた探索によって量子化値を決定できるので、より効率良く非線形な演算を行うことが可能となる。

【０１４６】

［実行結果］

ここで、図１５を参照して、上述した量子化処理を適用したときのサイクル数について説明する。図１５は、RISC(Reduced Instruction Set Computer) CPUであるMIPS社R4000を用いて、上述した量子化処理をそれぞれ実行したときのサイクル数を示している。

【０１４７】

図１５に示されるように、従来手法（式（１）に含まれるべき乗関数）の演算を含む量子化処理を実行したときのサイクル数を1.00とすると、条件文（範囲テーブル）を用いた（条件による探索を含む）量子化処理（図５）を実行したときのサイクル数は、0.28となり、72％の効率改善がされている。また、ハッシュテーブルを用いた量子化処理を実行したときのサイクル数は、0.32となり、68％の効率改善され、また、条件文とハッシュテーブルとを併用した量子化処理（図１２）を実行したときのサイクル数は、0.21となり、79％の効率改善されている。

【０１４８】

以上のように、本発明の量子化処理によれば、従来手法よりも効率を改善することができる。

【０１４９】

＜３．第３の実施の形態＞

［非線形関数と離散値］

以上においては、AAC規格の量子化処理について説明してきたが、ISO/IEC11172-3にて標準化されているMPEG-1オーディオレイヤ３（MP3）の量子化処理においても、式（１）および式（２）と同様の量子化および逆量子化処理が行われているので、MP3の量子化お

よび逆量子化処理においても、本発明を適用することができる。

【０１５０】

また、以上においては、非線形関数としてAAC規格の量子化処理で用いられるべき乗関数を例に説明してきたが、式（８）に示されるように、入力値Ｘに対する所定の非線形関数func(X)について、離散値Ｙを得る場合においても、本発明を適用することができる。

【０１５１】

【数７】

$$Y = (int)(func(X)) \quad \cdots (8)$$

【０１５２】

また、上述した説明では、離散値は整数であるものとして説明してきたが、式（９）に示されるように、入力値Ｘに対して、離散値Ｙが一意に定まればよく、その離散値Ｙが、浮動小数点数であっても、本発明を適用することができる。

【０１５３】

【数８】

$$Y = (int)(func(X)) + 0.45 \quad \cdots (9)$$

【０１５４】

さらに、上述したように、入力値Ｘに対して、離散値Ｙが一意に定まればよいので、その離散値Ｙをとる入力値Ｘの範囲は複数であってもよい。

【０１５５】

また、入力値Ｘを離散値Ｙに変換する演算処理の頻度の高い範囲にのみ、本実施の形態を適用し、その他の範囲に対しては、単に、式（８）で示されるような演算を行うようにしてもよい。

【０１５６】

さらに、以上においては、離散値Ｙをとる入力値Ｘの範囲を予め算出しておくようにしたが、例えば、入力値Ｘの離散値Ｙへの変換処理の最中に、離散値Ｙをとる入力値Ｘの範囲が変化する場合には、適宜、入力値Ｘの範囲を算出し直すようにしてもよい。

【０１５７】

［演算装置の構成例］

ここで、図１６のブロック図を参照して、入力値Ｘに対して、所定の非線形関数func(X)による演算を施し、離散値Ｙを出力する演算装置について説明する。

【０１５８】

図１３の演算装置４０１は、範囲テーブル作成部４３１および探索変換部４３２から構成される。

【０１５９】

範囲テーブル作成部４３１は、例えば、上述した式（７）において、入力値Ｘを順次変化させていき、離散値Ｙが変化したときの入力値Ｘと、そのときの離散値Ｙとを対応付けた範囲テーブルを作成し、探索変換部４３２に供給する。

【０１６０】

探索変換部４３２は、探索部４３２ａおよび決定部４３２ｂを備えている。探索部４３２ａは、範囲テーブル作成部４３１から供給された範囲テーブルにおいて、入力された入力値が含まれる範囲を探索する。決定部４３２ｂは、探索部４３２ａによって探索された範囲に対応付けられている離散値を決定し、外部の装置に出力する。

【０１６１】

［演算装置の範囲テーブル作成処理］

次に、図１７のフローチャートを参照して、図１６の演算装置４０１の範囲テーブル作成処理について説明する。演算装置４０１において、範囲テーブル作成処理は、離散値探索処理を行う前に行われる。

【０１６２】

ステップＳ２５１において、範囲テーブル作成部４３１は、例えば、上述した式（７）において、入力値Ｘを順次変化させていき、離散値Ｙが変化したときの入力値Ｘと、そのときの離散値Ｙとを対応付けた範囲テーブルを作成し、探索変換部４３２に供給する。

【０１６３】

以上の処理により、離散値出力処理を行う前に、入力値の範囲と、離散値とを対応付けた範囲テーブルを作成することができる。

【０１６４】

［演算装置の離散値探索処理］

次に、図１８のフローチャートを参照して、図１６の演算装置４０１の離散値探索処理について説明する。

【０１６５】

ステップＳ２７１において、探索変換部４３２の探索部４３２ａは、範囲テーブル作成部４３１から供給された範囲テーブルにおいて、入力された入力値が含まれる範囲を探索する。

【０１６６】

ステップＳ２７２において、探索変換部４３２の決定部４３２ｂは、探索部４３２ａによって探索された範囲に対応付けられている離散値を決定し、外部の装置に出力する。

【０１６７】

ステップＳ２７３において、探索変換部４３２は、全ての入力値に対して、離散値の探索の処理を終えたか否かを判定する。全ての入力値に対して、離散値の探索の処理を終えていないと判定された場合、処理は、ステップＳ２７１に戻り、全ての入力値に対して処理を終えるまで、ステップＳ２７１乃至Ｓ２７３を繰り返す。

【０１６８】

以上の処理により、予め求められている離散値に対応する入力値の範囲に基づいて、入力された入力値に対応する離散値を決定できる。したがって、例えば、非線形関数であるfunc(X)の演算をすることなく、範囲テーブルを用いた探索によって離散値を決定できるので、より効率良く非線形な演算処理を行うことが可能となる。

【０１６９】

なお、図１６の演算装置４０１は、１つの入力値Ｘに対して、その入力値が含まれる範囲と離散値Ｙとが対応付けられた、１つの範囲テーブルを有するようにしたが、入力値の種類毎に、それぞれの入力値の範囲と離散値とが対応付けられた、複数の範囲テーブルを有することができる。すなわち、演算装置４０１は、入力値の種類を示す情報やアドレス等に応じて、対応する範囲テーブルを読み出し、読み出した範囲テーブルを用いて、その入力値の範囲に対応する離散値を出力することができる。

【０１７０】

これにより、複数種類の入力値に対して、それぞれの離散値を出力する場合であっても、入力値の種類に応じた範囲テーブルを読み出すことによって、１つの演算装置のみで複数種類の離散値を出力することができる。

【０１７１】

上述した一連の処理は、ハードウェアにより実行することもできるし、ソフトウェアにより実行することもできる。一連の処理をソフトウェアにより実行する場合には、そのソフトウェアを構成するプログラムが、専用のハードウェアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータ等に、プログラム記録媒体からインストールされる。

【０１７２】

図１９は、上述した一連の処理をプログラムにより実行するコンピュータのハードウェアの構成例を示すブロック図である。

【０１７３】

コンピュータにおいて、CPU（Central Processing Unit）９０１，ROM（Read Only Mem

ory）９０２，RAM（Random Access Memory）９０３は、バス９０４により相互に接続されている。

【０１７４】

バス９０４には、さらに、入出力インタフェース９０５が接続されている。入出力インタフェース９０５には、キーボード、マウス、マイクロホン等よりなる入力部９０６、ディスプレイ、スピーカ等よりなる出力部９０７、ハードディスクや不揮発性のメモリ等よりなる記憶部９０８、ネットワークインタフェース等よりなる通信部９０９、磁気ディスク、光ディスク、光磁気ディスク、或いは半導体メモリ等のリムーバブルメディア９１１を駆動するドライブ９１０が接続されている。

【０１７５】

以上のように構成されるコンピュータでは、CPU９０１が、例えば、記憶部９０８に記憶されているプログラムを、入出力インタフェース９０５およびバス９０４を介して、RAM９０３にロードして実行することにより、上述した一連の処理が行われる。

【０１７６】

コンピュータ（CPU９０１）が実行するプログラムは、例えば、磁気ディスク（フレキシブルディスクを含む）、光ディスク（CD-ROM(Compact Disc-Read Only Memory),DVD(Digital Versatile Disc)等）、光磁気ディスク、もしくは半導体メモリ等よりなるパッケージメディアであるリムーバブルメディア９１１に記録して、あるいは、ローカルエリアネットワーク、インターネット、デジタル衛星放送といった、有線または無線の伝送媒体を介して提供される。

【０１７７】

そして、プログラムは、リムーバブルメディア９１１をドライブ９１０に装着することにより、入出力インタフェース９０５を介して、記憶部９０８にインストールすることができる。また、プログラムは、有線または無線の伝送媒体を介して、通信部９０９で受信し、記憶部９０８にインストールすることができる。その他、プログラムは、ROM９０２や記憶部９０８に、あらかじめインストールしておくことができる。

【０１７８】

なお、コンピュータが実行するプログラムは、本明細書で説明する順序に沿って時系列に処理が行われるプログラムであっても良いし、並列に、あるいは呼び出しが行われたとき等の必要なタイミングで処理が行われるプログラムであっても良い。

【０１７９】

また、本発明の実施の形態は、上述した実施の形態に限定されるものではなく、本発明の要旨を逸脱しない範囲において種々の変更が可能である。

【図面の簡単な説明】

【０１８０】

【図１】本発明を適用したオーディオ符号化装置の一実施の形態の構成例を示すブロック図である。

【図２】図１のオーディオ符号化装置における量子化部の構成例を示すブロック図である。

【図３】図２の量子化部の範囲テーブル作成処理について説明するフローチャートである。

【図４】範囲テーブルの例を示す図である。

【図５】図２の量子化部の量子化処理について説明するフローチャートである。

【図６】図５のフローチャートのステップＳ７２，Ｓ７３における処理をＣ言語で記述したプログラムの例を示す図である。

【図７】量子化値に対する量子化処理の頻度について説明する図である。

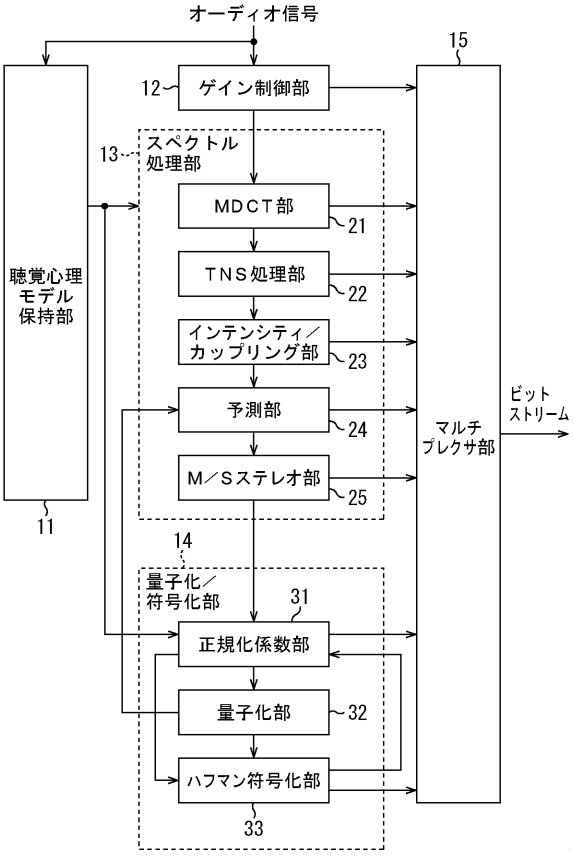
【図８】量子化部の他の構成例を示すブロック図である。

【図９】図８の量子化部のハッシュテーブル作成処理について説明するフローチャートである。

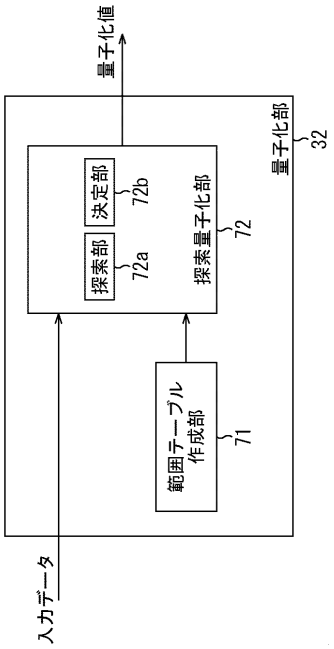
【図１０】浮動小数点型のデータについて説明する図である。

【図１１】ハッシュテーブルをＣ言語で記述した例を示す図である。
【図１２】図８の量子化部の量子化処理について説明するフローチャートである。
【図１３】図１２のフローチャートのステップＳ１７２乃至Ｓ１７８における処理をＣ言語で記述したプログラムの例を示す図である。
【図１４】図１３のプログラムの例において浮動小数点数を固定小数点数とした場合の例をＣ言語で記述したプログラムの例を示す図である。
【図１５】量子化処理を適用したときのサイクル数について説明する図である。
【図１６】本発明を適用した演算装置の一実施の形態の構成例を示すブロック図である。
【図１７】図１６の演算装置の範囲テーブル作成処理について説明するフローチャートである。
【図１８】図１６の演算装置の離散値探索処理について説明するフローチャートである。
【図１９】パーソナルコンピュータの構成例を示すブロック図である。
【符号の説明】
【０１８１】
３２　量子化部，　７１　範囲テーブル作成部，　７２　探索量子化部，　７２ａ　探索部，　７２ｂ　決定部，　１５１　量子化部，　１７１　ハッシュテーブル作成部，　１７２　探索量子化部，　１７３ａ　初期探索値決定部，　１７２ｂ　探索部，　１７２ｃ　決定部，　４０１　演算装置，　４３１　範囲テーブル作成部，　４３２　探索変換部，　４３２ａ　探索部，　４３２ｂ　決定部

【図１】
図1

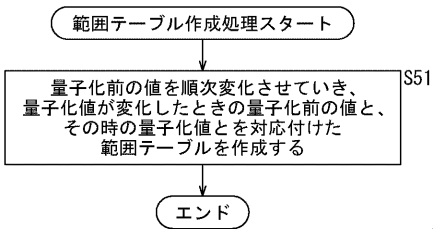


【図２】
図2



【図３】

図3



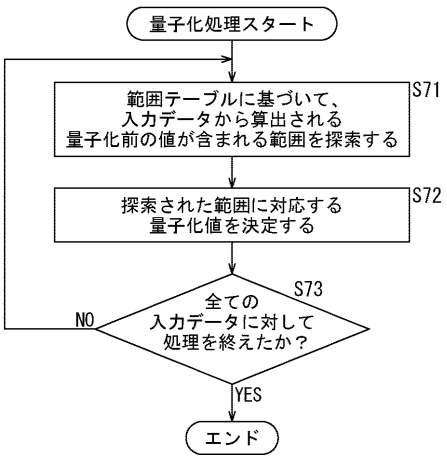
【図４】

図4

量子化値Z	量子化前の値Yの範囲
0	4.9999600e−01F
1	1.8629548e+00F
2	3.5652816e+00F
3	5.5063962e+00F
⋮	⋮
8189	1.6507573e+05F
8190	1.6510260e+05F
8191	1.6512946e+05F

【図５】

図5



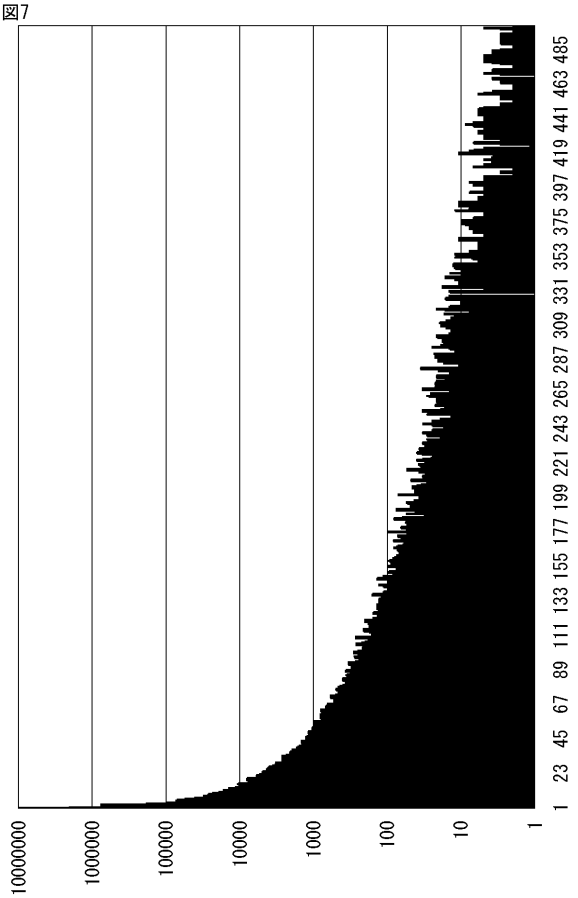
【図６】

図6

```
                                    81
                                    ⌇
 1       for (m = 0; m < data_size; m += 1) {
 2           if      (Y[m] < 4.9999600e-01F) {
 3               Z[m] =    0;
 4           }
 5           else if (Y[m] < 1.8629548e+00F) {
 6               Z[m] =    1;
 7           }
 8           else if (Y[m] < 3.5652816e+00F) {
 9               Z[m] =    2;
10           }
11           else if (Y[m] < 5.5063962e+00F) {
12               Z[m] =    3;
13           }
 ⋮
N-9          else if (Y[m] < 1.6507573e+05F) {
N-8              Z[m] = 8189;
N-7          }
N-6          else if (Y[m] < 1.6510260e+05F) {
N-5              Z[m] = 8190;
N-4          }
N-3          else if (Y[m] < 1.6512946e+05F) {
N-2              Z[m] = 8191;
N-1          }
N       }
```

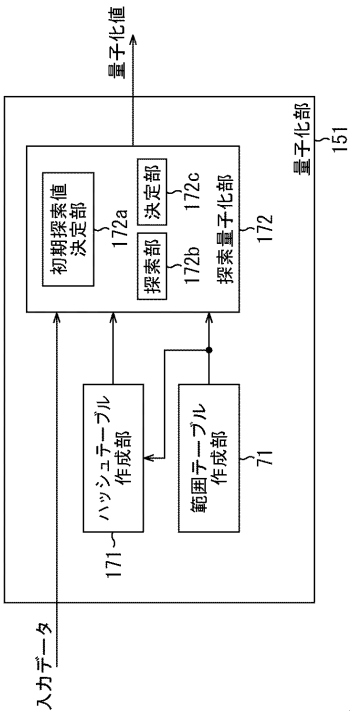
【図７】

図7



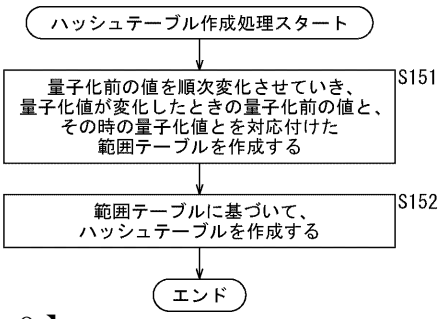
【図８】

図8



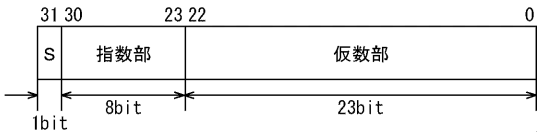
【図９】

図9



【図１０】

図10



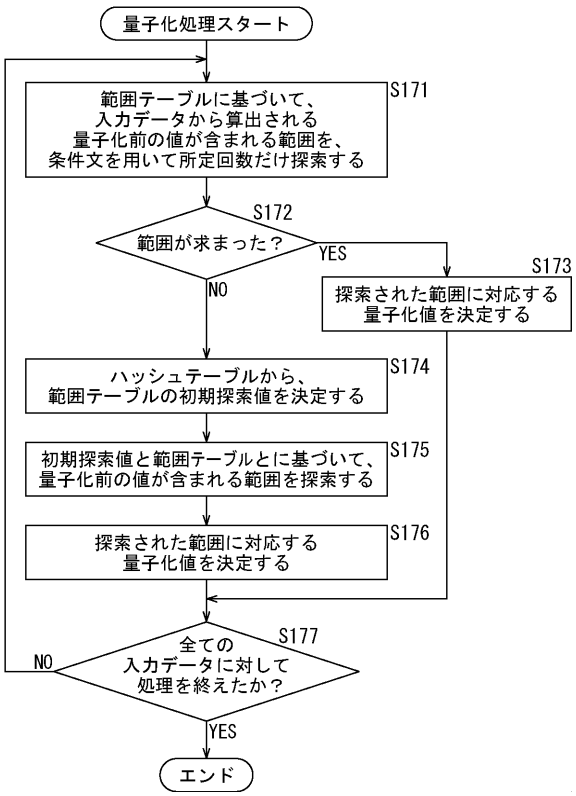
【図１１】

図11

```
181
 1      short quantize_hash[20][16]
 2      = {     /* 指数部index1 = 0 */
 3          {     0,    0,    0,    0,    0,    0,    0,    0,
 4                0,    0,    0,    0,    0,    0,    0,    0,
 5          },  /* 指数部index1 = 1 */
 6          {     0,    0,    0,    0,    0,    0,    0,    0,
 7                0,    0,    0,    0,    0,    0,    0,    0,
 8          },  /* 指数部index1 = 2 */
 9          {     0,    0,    0,    0,    0,    0,    0,    0,
10                0,    0,    0,    0,    0,    1,    1,    1,
11          },  /* 指数部index1 = 3 */
12          {     1,    1,    1,    1,    1,    1,    1,    1,
13                1,    1,    1,    1,    2,    2,    2,    2,
14          },  /* 指数部index1 = 4 */
15          {     2,    2,    2,    2,    2,    2,    3,    3,
16                3,    3,    3,    3,    3,    3,    4,    4,
17          },  /* 指数部index1 = 5 */
18          {     4,    4,    4,    5,    5,    5,    5,    5,
19                6,    6,    6,    6,    6,    7,    7,    7,
20          },  /* 指数部index1 = 6 */
21          {     7,    8,    8,    8,    9,    9,    9,   10,
22               10,   10,   11,   11,   11,   12,   12,   12,
23          },  /* 指数部index1 = 7 */
24          {    13,   14,   14,   15,   15,   16,   17,   17,
25               18,   18,   19,   19,   20,   20,   21,   22,
26          },  /* 指数部index1 = 8 */
27          {    23,   24,   25,   26,   27,   28,   29,   30,
28               31,   31,   32,   33,   34,   35,   36,   37,
29          } , /* 指数部index1 = 9 */
30          {    38,   40,   41,   43,   45,   47,   48,   50,
31               51,   53,   55,   56,   58,   59,   61,   62,
32          },
 :
N-7             /* 指数部index1 = 18 */
N-6         {  4096, 4286, 4474, 4659, 4842, 5023, 5201, 5377,
N-5            5552, 5724, 5895, 6064, 6232, 6398, 6563, 6726,
N-4         },  /* 指数部index1 = 19 */
N-3         {  6889, 7209, 7525, 7836, 8143, 8143, 8143, 8143,
N-2            8143, 8143, 8143, 8143, 8143, 8143, 8143, 8143,
N-1         },
N       }
```

【図１２】

図12



【図１３】

図13

191

```
1       float quantize_tab[8192]
2       = {
3           4.9999600e-01F, /*    0 */
4           1.8629548e+00F, /*    1 */
5           3.5652816e+00F, /*    2 */
6           5.5063962e+00F, /*    3 */
:
8192        1.6507573e+05F, /* 8189 */
8193        1.6510260e+05F, /* 8190 */
8194        1.6512946e+05F, /* 8191 */
8195    };
8196        union {
8197            float f;
8198            int   i;
8199        } uni0;
8200        for (m = 0; m < data_size; m += 1) {
8201            uni0.f = Y[m];
8202            if      (uni0.f < 4.9999600e-01F) {
8203                k = 0;
8204            }
8205            else if (uni0.f < 1.8629548e+00F) {
8206                k = 1;
8207            }
8208            else if (uni0.f < 3.5652816e+00F) {
8209                k = 2;
8210            }
8211            else if (uni0.f < 5.5063962e+00F) {
8212                k = 3;
8213            }
8214            else if (uni0.f < 7.6383047e+00F) {
8215                k = 4;
8216            }
8217            else {
8218            index1 = (uni0.i >> 23) - 125;              /* 指数部 */
8219            index2 = (uni0.i & 0x007FFFFF) >> (23 - 4); /* 仮数部 */
8220            k = quantize_hash[index1][index2];
8221            for ( ; k < 8192; k++) {
8222                if (uni0.f < quantize_tab[k]) {
8223                    break;
8224                }
8225            }
8226        }
8227        Z[m] = k;
8228    }
```

【図１４】

図14

201

```
1       int quantize_tab_int[8192]
2       = {
3               128, /*    0 */
4               486, /*    1 */
5               913, /*    2 */
6              1410, /*    3 */
:
8192        42259386, /* 8189 */
8193        42266266, /* 8190 */
8194        42273142, /* 8191 */
8195    };
8196        int yin;
8197        for (m = 0; m < data_size; m += 1) {
8198            yin = Y[m];
8199            if      (yin < 128) {
8200                k =    0;
8201            }
8202            else if (yin < 486) {
8203                k =    1;
8204            }
8205            else if (yin < 913) {
8206                k =    2;
8207            }
8208            else if (yin < 1410) {
8209                k =    3;
8210            }
8211            else if (yin < 1955) {
8212                k =    4;
8213            }
8214            else {
8215                mask = 0x40000000;
8216                for (cnt = 30; cnt >= 0; cnt--) {
8217                    if ((yin & mask > 0) {
8218                        break;
8219                    }
8220                    mask = mask >> 1;
8221                }
8222                yin = yin - mask;
8223                index1 = cnt - 8 + 2;      /* 指数部 */
8224                index2 = yin >> (cnt - 4); /* 仮数部 */
8225                k = quantize_hash[index1][index2];
8226                for ( ; k < 8192; k++) {
8227                    if (yin < quantize_tab_int[k]) {
8228                        break;
8229                    }
8230                }
8231            }
8232            Z[m] = k;
8233        }
```

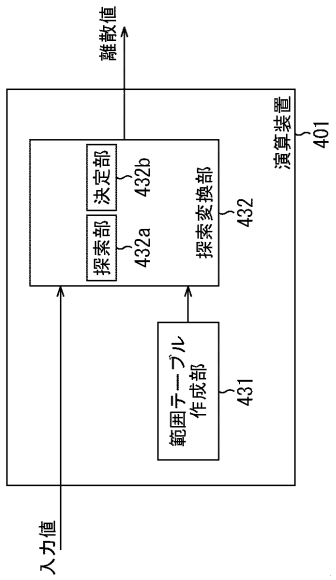
【図１５】

図15

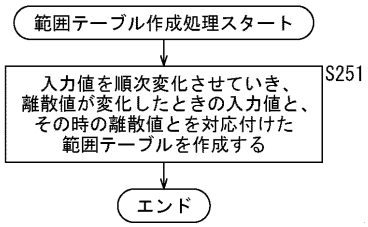
手法	比率
従来手法	1.00
条件文	0.28
ハッシュテーブル	0.32
条件文とハッシュテーブルの併用	0.21

【図１６】

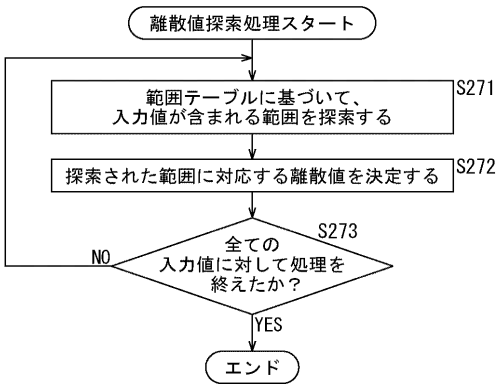
図16



【図１７】
図17



【図１８】
図18



【図１９】
図19