(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4717780号
(P4717780)
(45)発行日　平成23年7月6日(2011.7.6)　(24)登録日　平成23年4月8日(2011.4.8)

(51)Int.Cl.　　F I
H03M 7/40 (2006.01)　　H03M 7/40
H04N 7/30 (2006.01)　　H04N 7/133 Z

請求項の数 5　(全 18 頁)

(21)出願番号　特願2006-298214(P2006-298214)
(22)出願日　平成18年11月1日(2006.11.1)
(65)公開番号　特開2008-118307(P2008-118307A)
(43)公開日　平成20年5月22日(2008.5.22)
審査請求日　平成21年10月28日(2009.10.28)

(73)特許権者　000001007
キヤノン株式会社
東京都大田区下丸子３丁目３０番２号
(74)代理人　100076428
弁理士　大塚　康徳
(74)代理人　100112508
弁理士　高柳　司郎
(74)代理人　100115071
弁理士　大塚　康弘
(74)代理人　100116894
弁理士　木村　秀二
(72)発明者　野澤　慎吾
東京都大田区下丸子３丁目３０番２号　キヤノン株式会社内

最終頁に続く

(54)【発明の名称】符号化装置及びその制御方法

(57)【特許請求の範囲】
【請求項１】
符号化対象の２値シンボルを順に入力し、０以上１未満の小数点以下の値を示す２進データを符号化データとして生成し、生成した符号化データを出力する算術符号化を用いた符号化装置であって、
２値シンボルを入力する度に、有限精度で表現される数直線上の区間を、優勢シンボルと劣勢シンボルの発生確率の比に従って２分割すると共に、入力した２値シンボルの値に従って、分割した一方の区間を選択することを繰り返す区間分割／選択手段と、
該区間分割／選択手段で選択された選択区間の長さが、予め設定された区間長を下回る場合に、前記選択区間の長さを拡大するための正規化を行なう正規化手段と、
該正規化手段の正規化対象の前記選択区間の上端と下端の値を判定する上下端判定手段と、
前記正規化手段による前記選択区間の正規化を行なう際に、前記上下端判定手段で判定された上端と下端の値に応じて、符号化データのビットの値の決定と出力、又は、出力すべきビットの保留個数の増加を行なう処理手段とを備え、
前記処理手段は、
前記正規化対象となる前記選択区間内に、注目する小数の桁を決定するための閾値が存在するか否かを判定する判定手段と、
該判定手段で、前記選択区間内に前記閾値が存在すると判定した場合、保留ビット数を１つ増加させる保留ビット増加手段と、

前記判定手段で、前記選択区間外に前記閾値が存在すると判定した場合、前記選択区間と前記閾値との相対位置に基づいて決定される１ビットのデータを出力し、保留ビット数の数分のビットのデータを決定し、出力すると共に、保留ビット数をゼロクリアする符号ビット出力手段とを含む
ことを特徴とする符号化装置。
【請求項２】
更に、画像データを入力する画像データ入力手段と、
入力した画像データを周波数変換する周波数変換手段と、
周波数変換して得られた係数データを量子化する量子化手段と、
該量子化手段による量子化後の係数データを２値化し、前記２値シンボルとして出力する２値化手段と
を備えることを特徴とする請求項１に記載の符号化装置。
【請求項３】
符号化対象の２値シンボルを順に入力し、０以上１未満の小数点以下の値を示す２進データを符号化データとして生成し、生成した符号化データを出力する算術符号化を用いた符号化装置の制御方法であって、
２値シンボルを入力する度に、有限精度で表現される数直線上の区間を、優勢シンボルと劣勢シンボルの発生確率の比に従って２分割すると共に、入力した２値シンボルの値に従って、分割した一方の区間を選択することを繰り返す区間分割／選択工程と、
該区間分割／選択工程で選択された選択区間の長さが、予め設定された区間長を下回る場合に、前記選択区間の長さを拡大するための正規化を行なう正規化工程と、
該正規化工程による正規化対象の前記選択区間の上端と下端の値を判定する上下端判定工程と、
前記正規化工程による前記選択区間の正規化を行なう際に、前記上下端判定工程で判定された上端と下端の値に応じて、符号化データのビットの値の決定と出力、又は、出力すべきビットの保留個数の増加を行なう処理工程とを備え、
前記処理工程は、
前記正規化対象となる前記選択区間内に、注目する小数の桁を決定するための閾値が存在するか否かを判定する判定工程と、
該判定工程で、前記選択区間内に前記閾値が存在すると判定した場合、保留ビット数を１つ増加させる保留ビット増加工程と、
前記判定工程で、前記選択区間外に前記閾値が存在すると判定した場合、前記選択区間と前記閾値との相対位置に基づいて決定される１ビットのデータを出力し、保留ビット数の数分のビットのデータを決定し、出力すると共に、保留ビット数をゼロクリアする符号ビット出力工程とを含む
ことを特徴とする符号化装置の制御方法。
【請求項４】
コンピュータが読込み実行することで、前記コンピュータを、請求項１又は２に記載の各手段として機能させることを特徴とするコンピュータプログラム。
【請求項５】
請求項４に記載のコンピュータプログラムを格納したことを特徴とするコンピュータ可読記憶媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、画像データ等のデジタル情報から生成された２値シンボル列を算術符号化する技術に関するものである。
【背景技術】
【０００２】
近年、デジタル信号処理技術の進歩により、動画像や静止画像、音声等、大量のデジタ

ル情報を高能率符号化し、小型記録媒体への記録や通信媒体による伝送を行うことが可能になっている。このような技術を応用し、テレビ放送やビデオカメラの映像をストリームに変換できる映像符号化装置の開発が行われている。動画像の映像符号化方法の中では、とりわけＩＴＵ－Ｔが策定した規格Ｈ．２６４（別名ＭＰＥＧ４ Ｐａｒｔ１０／ＡＶＣ）が注目されている。昨今における高能率符号化の圧縮効果は、エントロピー符号化によるものが大きく、特に効率の高いエントロピー符号化方法として算術符号化が挙げられ、上記のＨ．２６４においても採用されている。

【０００３】

Ｈ．２６４ではContext－based Adaptive Binary Arithmetic Codingと呼ばれる算術符号化（以下、ＣＡＢＡＣと称す）と、Context－based Adaptive Variable length Codingと呼ばれる可変長符号化（以下、ＣＡＶＬＣと称す）とが採用されている。

【０００４】

前記ＣＡＢＡＣ及びＣＡＶＬＣに着目した先行例として、特許文献１が提案されている。この特許文献１の「画像情報符号化方法及び画像情報復号方法」には、ＣＡＢＡＣへの入出力データ量を制限し、復号化器の処理時間を保証することを目的とした発明が開示されている。

【特許文献１】特開２００４－１３５２５１号公報

【発明の開示】

【発明が解決しようとする課題】

【０００５】

図５は従来の算術符号化方法の処理手順を示すフローチャートである。なお、この処理手順はＩＴＵ－Ｔ Ｈ．２６４規格書により定められており、公知のものである。Ｈ．２６４における算術符号化、すなわちＣＡＢＡＣは、２値シンボルの入力に対応している。

【０００６】

図５においてステップＳ９００でシンボルの入力が終了したか否かを判定し、未完であると判断した場合にはステップＳ９０１にてシンボルを入力する。ここで入力されるシンボルとは０または１の２値シンボルである。次にステップＳ９０２は区間を分割する。区間とは、０から１０２３までの整数領域内で下端と長さによって表現される。区間の分割とは、２値シンボルの発生確率の比により、区間を二つに分けることである。例えば、シンボル“０”の発生確率が７５％で、シンボル“１”の発生確率が２５％の場合、区間は７５：２５の比で分割する（この場合、シンボル“０”を優勢シンボル、シンボル“１”を劣勢シンボルと呼ぶ）。続くステップＳ９０３は、前記入力シンボルに従って、この分割された区間のいずれか一方を選択する。

【０００７】

例えば、図６のように、現在の区間が下端の値が２１、長さ３２０で表されているとする。０と１の発生確率が７５％と２５％の場合とする。今、入力したシンボルが“１”であるとき、“１”の区間が選択される。この結果、新たな下端は２６１、長さは８０となる。

【０００８】

次にステップＳ９０４では、新たな区間の長さが２５６未満であるか否かを判定する。区間の長さが２５６以上の場合は、ステップＳ９００へと戻る。また、図６で説明した例のように区間の長さ８０となり、２５６を下回る場合は、ここでステップＳ９０５へ分岐する。

【０００９】

ステップＳ９０５では、区間の下端の値が２５６未満であるか否かを判定する。区間の下端が２５６未満の場合はステップＳ９０６へ分岐し、符号列として“０”を出力し、ステップＳ９０７は後述する保留ビット数に相当する個数の“１”を符号列として出力する。また、保留ビット数はここでゼロにリセットされる。ステップＳ９０８は、区間の正規化を行う。正規化とは、２５６未満になった区間の長さを拡大し、以降の処理の精度を確保する処理である。このステップＳ９０８では、下端の値と区間の長さをそれぞれ２倍に

する。
【００１０】
ステップＳ９０５において、下端が２５６以上であると判定した場合、処理はステップＳ９０９に分岐し、区間の下端が５１２未満か否かを判定する。区間の下端が５１２未満の場合にはステップＳ９１０へ、５１２以上の場合にはステップＳ９１２へ分岐する。
【００１１】
ステップＳ９１２では、符号列として“１”を出力し、ステップＳ９１３では後述する保留ビット数に相当する個数の“０”を続く符号列として出力する。保留ビット数はここでもゼロにリセットされる。ステップＳ９１４では、区間の正規化を行う。このステップによる正規化では、下端から５１２を引いて２倍した値を新たな下端とし、また、区間の長さを２倍する。
【００１２】
また、ステップＳ９０９において、下端が５１２を下回る場合はステップＳ９１０へ分岐し、保留ビット数を１つ増加させる。続いてステップＳ９１１では、区間の正規化を行う。このステップＳ９１１による正規化は、下端から２５６を引き２倍、区間も２倍する。例えば図６の例では、下端が１０になり（＝（２６１－２５６）×２）、区間の長さは１６０（＝８０×２）。また、保留ビットも“１”に増加する。これら正規化処理の各ステップの後、再度ステップＳ９０４へ戻って、区間の長さが２５６以上になるまで、同様に正規化処理を繰り返す。
【００１３】
例えば図６の例では、正規化後の区間の長さは１６０なので、再度ステップＳ９０４からステップＳ９０５へ進む。下端が１０なので、ステップＳ９０６へと進み、“０”を出力、先ほど保留ビット数を１に増加しているので、１個の“１”を出力する（ステップＳ９０７）。すなわち符号列“０１”が出力される。さらに正規化のステップＳ９０８により、下端を２倍の２０、区間の長さも２倍の３２０とする。ここで区間の長さは２５６以上となるので、ステップＳ９０４からＳ９００へと戻る。このステップＳ９００により入力シンボルの終了を検知し、ステップＳ９１５の終了処理ステップへ分岐するまで上記処理を繰り返す。
【００１４】
本発明者は、以上の従来の算術符号化方法には、２つの問題点があると考えた。第１には、ステップＳ９０４からＳ９１４までのループにより、ひとつの入力シンボルを処理するためのステップ数が非常に多いことである。これは、ソフトウェアによる処理時間の増大や、ハードウェアによる回路規模を増大の要因となる。第２には、ステップＳ９１０における、保留ビット数の増加が繰り返されると、その値の保持に必要なメモリが多くなり、更に保留ビットの解消処理であるステップＳ９０７、Ｓ９１３の処理回数も増大する点である。
【００１５】
本発明はかかる問題に鑑みなされたものであり、算術符号化をハードウェアに適用するのであれば、回路を簡略化できて消費電力を少なくすることを可能にし、ソフトウェアに適用するのであれば、処理の簡略化に伴うメモリ消費量の削減と高速処理を可能ならしめる技術を提供することにある。
【課題を解決するための手段】
【００１６】
かかる課題を解決するため、例えば第１の発明の符号化装置は以下の構成を備える。すなわち、
符号化対象の２値シンボルを順に入力し、０以上１未満の小数点以下の値を示す２進データを符号化データとして生成し、生成した符号化データを出力する算術符号化を用いた符号化装置であって、
２値シンボルを入力する度に、有限精度で表現される数直線上の区間を、優勢シンボルと劣勢シンボルの発生確率の比に従って２分割すると共に、入力した２値シンボルの値に

従って、分割した一方の区間を選択することを繰り返す区間分割／選択手段と、
該区間分割／選択手段で選択された選択区間の長さが、予め設定された区間長を下回る場合に、前記選択区間の長さを拡大するための正規化を行なう正規化手段と、
該正規化手段の正規化対象の前記選択区間の上端と下端の値を判定する上下端判定手段と、
前記正規化手段による前記選択区間の正規化を行なう際に、前記上下端判定手段で判定された上端と下端の値に応じて、符号化データのビットの値の決定と出力、又は、出力すべきビットの保留個数の増加を行なう処理手段とを備え、
前記処理手段は、
前記正規化対象となる前記選択区間内に、注目する小数の桁を決定するための閾値が存在するか否かを判定する判定手段と、
該判定手段で、前記選択区間内に前記閾値が存在すると判定した場合、保留ビット数を１つ増加させる保留ビット増加手段と、
前記判定手段で、前記選択区間外に前記閾値が存在すると判定した場合、前記選択区間と前記閾値との相対位置に基づいて決定される１ビットのデータを出力し、保留ビット数の数分のビットのデータを決定し、出力すると共に、保留ビット数をゼロクリアする符号ビット出力手段とを含むことを特徴とする。

【発明の効果】

【００１８】

第１の発明によれば、保留ビット数を増加する処理の回数がこれまでの技術よりも少なくできるので、ハードウェアに適用するのであれば、回路を簡略化できて消費電力を少なくすることを可能になる。また、ソフトウェアに適用するのであれば、処理の簡略化に伴うメモリ消費量の削減と処理を更に高速なものとすることが可能になる。

【発明を実施するための最良の形態】

【００２０】

以下、添付図面に従って本発明に係る実施形態を詳細に説明する。

【００２１】

＜第１の実施形態＞

図７は、本実施形態による装置の構成例を示すブロック図である。１０１は映像入力部、１０２は変換部、１０３は量子化部、１０４はＣＡＶＬＣ符号化部、１０５はＣＡＢＡＣ符号化部、１０６は２値化部、１０７は算術符号化部、１０８は選択部、１０９はストリーム出力部である。本実施形態における符号化装置は、Ｈ．２６４規格に基づき、以下の手順で高能率符号化処理を行う。なお、本符号化装置は撮像装置（ビデオカメラ）や映像記録再生装置に適用可能な構成である。

【００２２】

映像入力部１０１から入力されるデジタル映像信号は、変換部１０２に供給される。変換部１０２は、入力したデジタル映像信号で表わされる画像データを、例えば１６×１６画素で構成されるブロックに分割し、各ブロック毎に係数列へ変換する。この変換部１０２は、動き予測処理、動き補償処理及び直交変換処理（周波数変換）などにより、ブロックにおける情報量の視覚的な冗長性を削減する。

【００２３】

変換部１０２の変換処理で得られた係数列は、量子化部１０３へ供給される。量子化部１０３は、入力した係数列を、予め設定された量子化パラメータに従って量子化する。

【００２４】

この量子化部１０３における量子化パラメータの大小に応じて、係数列の情報量は削減される。量子化された係数列は、ＣＡＶＬＣ符号化部１０４とＣＡＢＡＣ符号化部１０５に供給される。ＣＡＶＬＣ符号化部１０４については、本発明に直接関係が無いので詳細な説明は省略し、ＣＡＢＡＣ符号化部１０５について説明する。

【００２５】

ＣＡＢＡＣ符号化部１０５内の２値化部１０６は、入力した係数列を０、１の２値シン

ボル列へ変換する。

【００２６】

算術符号化部１０７は、２値化部１０６で２値化された２値シンボル列の順次を入力し、後述する手順に従った処理を行なうことで、圧縮した算術符号化データ列を生成し出力する。この時、２値化部１０６はテーブルによって各係数を２値シンボル列へ１対１に変換できるので、処理は１係数単位に実行することができる。一方、算術符号化部１０７は、２値シンボル１つずつ処理を行う必要がある。

【００２７】

ＣＡＶＬＣ符号化部１０４によって発生した符号列と、算術符号化が行われたＣＡＢＡＣ符号化部１０５によって発生した符合列は、それぞれ選択部１０８に供給され、選択部１０８によっていずれか一方の符号列（ストリーム）が出力として選択される。その後、選択されたストリームがストリーム出力部１０９より出力される。

【００２８】

本実施形態における特徴は、図７における算術符号化部１０７にある。そこで、この算術符号化部１０７を更に詳しく説明する。

【００２９】

算術符号化は、既に説明したように、入力されるシンボルの値に応じて区間の分割、選択を繰り返し、０以上の１未満の小数点の値を符号化結果とするものである。０以上の１未満の小数点の値は、「０．ｘｘｘ…（ｘは０または１）」の形式となるので、整数部分と小数点を除いた「ｘｘｘ…」が符号化データとなる。小数点第１位のビットは、十進数の“０．５”を示し、小数点第２位のビットは“０．２５”を示す。つまり、小数点第ｉ桁のビットで表わされる値は、小数点第ｉ－１桁のビットで表わされる値の半分と言える。例えば、十進数の“０．７５”は２進数では“０．１１”と表現できる。

【００３０】

また、電子回路、又は、コンピュータプログラムで処理するわけであるから、区間の下端、上端、区間長さは限られたビット数のレジスタで表現しなければならない。ところが、各小数点以下の桁のビットを求めていくと、それに応じて区間の長さは次第に狭くなっていく。つまり、区間の下端と上端を定義するレジスタの値の差は次第に小さくなっていき、シンボルの発生確率の比で区間を分割する際の精度が低くなっていく。それゆえ、使用するビット数を有効に機能するようにするため、区間の長さある閾値未満となったとき、その区間の長さを拡大する処理が必要になる。この区間の長さの拡大処理を正規化と言う。

【００３１】

図１は、本実施形態における算術符号化部１０７における算術符号化処理手順を示すフローチャートである。以下、このフローチャートに従って算術符号化部１０７の処理内容を説明する。

【００３２】

先ず、ステップＳ１００では、符号化すべきシンボルの入力が終了したか否かを判断する。この判断は、入力したシンボルの個数を計数し、その個数が予め定められた符号化単位であるシンボル数になったか否かで判断すればよい。

【００３３】

符号化処理が未完であると判断した場合には、ステップＳ１０１に処理を進め、シンボルを１つ入力する。本実施形態での符号化対象のシンボルは、２値化部１０６から出力される０または１の２値シンボルである。

【００３４】

次にステップＳ１０２では、区間を分割する。符号化の開始時の区間は０乃至１０２３の範囲である。すなわち、下端が０、区間の長さが１０２４であり、下端を０、上端の１０２３を整数の“１”と見立て、その間が小数点以下を表わすものとして定義する。この場合、下端と上端はそれぞれ１０ビットのレジスタで記憶保持すれば良いことになる。

【００３５】

ステップＳ１０２の区間の分割処理とは、入力した２値シンボルの発生確率の比により、区間を２つに分けることである。例えば、０の発生確率が７５％で、１の発生確率が２５％の場合、区間は７５：２５の比で分割される。

【００３６】

続くステップＳ１０３では、入力シンボルに従って、この分割された区間のいずれか一方を選択する。例えば、図６のように、現在の区間の下端が“２１”、その区間の長さが“３２０”であり、シンボル０と１の発生確率が７５％、２５％であるとする。今、入力したシンボルが“１”であった場合、１の区間が選択され、新たな下端は“２６１”、その区間の長さは“８０”となる。

【００３７】

次にステップＳ１０４は、新たな区間の長さを閾値“２５６”と比較する。区間の長さが“２５６”以上の場合は、ステップＳ１００へと戻る。

【００３８】

また、図６に示すように、区間の長さが“８０”となって、“２５６”を下回る場合、処理はステップＳ１０５に分岐する。

【００３９】

ステップＳ１０５では、区間の下端の値を判定する。下端の値が“５１２”に満たない場合、処理はステップＳ１０６に分岐する。このステップＳ１０６では、区間の上端の値が“５１２”未満であるか否かを判定する。

【００４０】

ステップＳ１０６にて、上端の値が“５１２”未満であると判定した場合、ステップＳ１０７に処理を進める。

【００４１】

ステップＳ１０７に処理を進むのは、選択した区間の上下端の値が共に“５１２”以下の場合である。従って、ステップＳ１０７では、“０”を符号化後のビットとして出力する。そして、続くステップＳ１０８では、その時点での保留ビット数で示される数だけ“１”を出力した後、保留ビット数を“０”にリセットする。なお、ステップＳ１０８に処理が進んだ直後の保留ビット数が“０”の場合には、何もしない。そして、ステップＳ１０９にて、区間の正規化を行う。このステップＳ１０９での正規化処理とは、“２５６”未満となった区間を拡大し、以降の符号化処理の精度を確保する処理である。このステップＳ１０９では、下端の値を２倍にした値を更新後の区間の下端の値とする。また、区間の長さを２倍にし、その結果を更新後の区間の長さとする。そして、ステップＳ１０４に戻る。

【００４２】

一方、ステップＳ１０５にて、選択区間の下端の値が“５１２”以上であると判定した場合、処理はステップＳ１１２に分岐する。すなわち、現在の区間の上端及び下端の値が共に、“５１２”以上である場合である。ステップＳ１１２では、注目桁の値として“１”を符号化後のビットとして出力する。続くステップＳ１１３では、保留ビット数で示される数の“０”を出力した後、保留ビット数を“０”にリセットする。そして、ステップＳ１１４にて、区間の正規化を行う。このステップＳ１１４での正規化処理とは、“２５６”未満となった区間を拡大し、以降の符号化処理の精度を確保する処理である。このステップＳ１１４では、下端の値から５１２を減じ、その減算結果を２倍にした値を更新後の下端の値にする。また、区間の長さを２倍した結果を、更新後の区間の長さとする。この後、処理はステップＳ１０４に戻る。

【００４３】

また、ステップＳ１０６にて、区間の上端の値が“５１２”以上であると判断した場合、処理はステップＳ１１０に進む。ステップＳ１１０に進むのは、区間の下端の値が“５１２”未満であり、区間の上端の値が“５１２”以上の場合である。つまり、注目桁のビットが“０”か、“１”かは、現時点では未定であることを示している。それ故、ステップＳ１１０にて、保留ビット数を１つ増加させる。続いて、ステップＳ１１１にて、区間

の正規化を行う。このステップＳ１１１における正規化は、区間の下端の値から“２５６”を減じ、その減算結果を２倍にした値を更新後の下端の値とする。また、区間の長さを２倍にした結果を更新後の区間の長さとする処理を行なう。この後、処理はステップＳ１０４に戻る。

【００４４】

こうして、符号化すべき全シンボルについてステップＳ１００乃至Ｓ１１４の処理を終えると、ステップＳ１１５の終了処理を行なう。この終了処理では、直前までの処理で決定された区間の上端、下端の間にあって、二進表記で都合の良い値を１つ決定し、その決定した値に従って、未出力の桁のビットを出力する処理を行なう。例えば、区間の上端の値が、区間の下端の値と等しくなったものとし（区間の長さが０になったものとし）、その際の符号化ビット及び保留ビットを出力する。

【００４５】

以上、本実施形態の算術符号化部１０７の処理内容を説明した。本実施形態の処理は、先に説明した図５の前述した従来方法よりも、保留ビット数を増加させるステップの実行回数を半減させることが可能になる。

【００４６】

例えば、図６の例では、区間選択により得られた新しい区間の下端の値が“２６１”である。従来方法では、ステップＳ９０５、Ｓ９０９による分岐で、保留ビット数を増加させるステップＳ９１０へと進む。

【００４７】

これに対し、本実施形態によれば、区間の下端の値が“２６１”で、尚且つ、区間の上端の値が“３４１”（＝２６１＋８０）であるから、ステップＳ１０５、１０６の分岐処理で、ステップＳ１０７以降の処理を行なう。すなわち、上記の場合、本実施形態によれば、保留ビットを増加する処理は行なわない。

【００４８】

また、ステップＳ１０７では、符号化データとしてビット“０”を出力する。この後、処理はステップＳ１０８に進むが、図６の場合、保留ビット数がゼロのままである。従って、このステップＳ１０８は無処理となる。そして、ステップＳ１０９の正規化では、現在の区間の下端の値と区間の長さをそれぞれ２倍にする。すなわち、区間の下端の値は“５２２”、区間の長さが“１６０”になる。この後、ステップＳ１０４に戻るが、区間の長さ“１６０”は、閾値“２５６”未満であるので、ステップＳ１０４、Ｓ１０５による分岐処理を経たのち、ステップＳ１１２に処理が進む。ステップＳ１１２では、“１”を出力し、ステップＳ１１３は無処理となる。そして、ステップＳ１１４の正規化処理の結果、区間の下端の値は“２０”（＝（５２２－５１２）×２）、区間の長さは“３２０”（＝１６０×２）となる。

【００４９】

以上により、出力される符号化データは“０１”となり、区間の下端の値は“２０”、区間の長さは“３２０”となる。ここで注目すべき点は、本実施形態の場合、保留ビットの増加処理を行なっていないにもかかわらず、図５に示した従来方法と同じ出力、同じ区間状態になっていることである。

【００５０】

ここで、従来技術と本実施形態の処理の違いを図２を用いて説明する。図２は正規化前の区間を上端と下端の範囲で分類したベン図である。

【００５１】

従来方法では、区間の長さが“２５６”未満となり、符号化データとしてビット“０”を出力するのは、その区間が下端の値が“２５６”未満の領域２０１のみである。すなわち、残る領域２０２、２０３では保留ビットを増加させる処理を行なわざるをえない。

【００５２】

一方、本実施形態による方法では、区間の長さが“２５６”未満であるとき、区間が領域２０１、２０２の条件を満たすとき、符号化データとしてビット“０”を出力する。そ

して、保留ビットを増加させるのは、領域２０３、すなわち、区間の上端の値が“５１２”以上であり、区間の下端の値が“５１２”未満の場合のみである。

【００５３】

このように本実施形態の如く、分類に上端を取り入れて詳細化することで、区間が本来保留する必要の無い領域２０２にあるときに、“０”を符号化データとして出力することで、保留ビット増加の処理の回数は、従来技術の半分まで減らすことが可能になる。因に、区間の下端の値が“５１２”以上を示す領域２０４では、符号化データのビットとして“１”を出力する。従って、区間の下端の値が“５１２”以上を示す領域２０４における、本実施形態の処理は、従来技術と方法と一致する。

【００５４】

図８は、実施形態における算術符号化部１０７の具体的な内部のブロック構成図である。

【００５５】

本実施形態の算術符号化部１０７は、図示の如く、入力部８０１、制御部８０２、終了処理部８０３、出力切替部８０４、区間更新部８０５、正規化部８０６、出力ビット生成部８０７、保留ビット処理部８０８で構成される。この構成における動作を、再び、図１のフローチャートと対応づけて説明する。

【００５６】

入力部８０１は、ステップＳ１０１にて、２値化部１０６から２値シンボルを順次入力する。また、入力部８０１は、入力したシンボル数が符号化単位の予め設定された個数になったことを検知すると、終了判定信号を制御部８０２に出力する。

【００５７】

区間更新部８０５はステップＳ１０２による区間の分割とステップＳ１０３における区間の選択を行い、選択された区間の上端と下端の値を示す信号を制御部８０２と正規化部８０６へ供給する。

【００５８】

制御部８０２は各ブロックから供給される信号に基づき、制御信号を各ブロックへ供給する。

【００５９】

終了処理部８０３は、制御部からのトリガー信号により、ステップＳ１１５における終了処理を行い、終了ビット列を出力切替部８０４に供給する。

【００６０】

出力ビット生成部８０７は制御部８０２からの制御信号に従い、“０”もしくは“１”を出力切替部８０４へ供給する。この処理は、図１におけるステップＳ１０７もしくはステップＳ１１２に相当する。

【００６１】

保留ビット処理部８０８は制御部８０２からの制御信号に従い、ステップＳ１０８、ステップＳ１１０、ステップＳ１１３における保留ビット列を出力切替部８０４へ供給する。

【００６２】

出力切替部８０４は制御部８０２からの制御信号に従い、終了処理部８０３、出力ビット生成部８０７、保留ビット処理部８０８から供給されるビット列のいずれかを選択し出力する。

【００６３】

正規化部８０６は制御部８０２からの制御信号と、区間更新部からの上端と下端の値を表す信号を受け取り、区間を正規化して新たな区間の上端と下端を区間更新部８０５へ供給する。この処理は、図１のステップＳ１０９、Ｓ１１１、Ｓ１１４に相当するものであるが、いずれの処理を行なうかは、制御部８０２からの制御信号によって決定される。正規化により変更された上端と下端の値は、区間更新部８０５を経由して、制御部８０２へ供給される。

【００６４】
制御部８０２は、図１のフローチャートの分岐処理であるステップＳ１００、Ｓ１０４、Ｓ１０５、Ｓ１０６の判断に依存し、これら各ブロックを制御するための信号を生成することになる。
【００６５】
以上説明したように本実施形態によれば、２値シンボル列を符号化する際に、保留ビット数を増加する処理回数を、これまでの半分に減らすことが可能になり、尚且つ、生成される符号化データのコンパチビリティも保証することが可能になる。従って、ハードウェアによる回路規模の増大を防ぐ事が可能になる。
【００６６】
また、符号化対象の画像データを入力する入力装置（ビデオカメラ、イメージスキャナ、或いは、非圧縮の画像データを蓄積する蓄積装置）を有し、それ以外の図７に示す各構成要素をマイクロコンピュータ等の高度な制御部と、その制御部が実行するプログラムを記憶したＲＡＭによって構成することも勿論可能である。また、図７における算術符号化部１０７のみ、マイクロコンピュータとそれが実行するソフトウェアによって実現させても構わない。
【００６７】
＜第２の実施形態＞
本発明に係る第２の実施形態を説明する。図３は、本第２の実施形態における算術符号化部１０７の処理手順を示すフローチャートである。算術符号化部１０７以外は、第１の実施形態と同じとし、その説明は省略する。
【００６８】
以下、図３のフローチャートに従って、算術符号化部１０７の処理手順を説明する。
【００６９】
先ず、ステップＳ３００では、符号化すべきシンボルの入力が終了したか否かを判断する。この判断は、入力したシンボルの個数を計数し、その個数が予め定められた符号化単位であるシンボル数になったか否かで判断すればよい。
【００７０】
符号化処理が未完であると判断した場合には、ステップＳ３０１に処理を進め、シンボルを１つ入力する。本実施形態での符号化対象のシンボルは、２値化部１０６から出力される０または１の２値シンボルである。
【００７１】
次にステップＳ３０２では、区間を分割する。符号化の開始時には、区間の下端が０、区間の長さが１０２４である。
【００７２】
ステップＳ３０２の区間の分割処理とは、入力した２値シンボルの発生確率の比により、区間を２つに分けることである。例えば、０の発生確率が７５％で、１の発生確率が２５％の場合、区間は７５：２５の比で分割される。続く、ステップＳ３０３では、入力シンボルに従って、この分割された区間のいずれか一方を選択する。例えば、図６のように、区間が下端２１と長さ３２０で表されていて、シンボル“０”と“１”の発生確率が７５％と２５％の場合とする。この場合、入力シンボルが“１”であった場合には、１の区間が選択され、新たな区間の下端は“２６１”、長さは“８０”となる。
【００７３】
次にステップＳ３０４では、ステップＳ３０３で選択した新たな区間の長さによって、正規化回数を算出する。既に述べたように、正規化処理は、区間の長さが２５６未満の時に発生し、区間の長さを２倍にする処理である。そして、正規化処理は、区間の長さが２５６以上となるまで繰り返される。
【００７４】
即ち、区間の長さと正規化回数との関係は次の通りとなる。
・区間の長さ１２８以上、２５６未満の場合は正規化回数は１回、

・区間の長さが６４以上、１２８未満の場合には正規化回数は２回、
・区間の長さが３２以上、６４未満の場合には正規化回数は３回、
・区間の長さが１６以上、３２未満の場合には正規化回数は４回、
・区間の長さが８以上、１６未満の場合には正規化回数は５回、
・区間の長さが４以上、８未満の場合には正規化回数は６回。

【００７５】

上記を考察すると、区間の長さを９ビットで２進表現した場合、正規化回数は、最上位ビット（ＭＳＢ）から最下位ビット（ＬＳＢ）に向かうゼロの連続数と一致する。従って、ステップＳ３０４では、ＭＳＢからゼロの連続数をカウントし、そのカウントした値を正規化回数として求める。なお、正規化回数を求める方法は上記に限らない。例えば、２を低とする区間の長さの対数を求め、その対数の小数点以下を切り捨てる。そして、切り捨てた対数を、８から減じた値を、正規化回数としても構わない。ただし、前者の方が、単純な処理、又は、単純な回路構成で実現できる。

【００７６】

続いてステップＳ３０５では、確定して出力する符号ビット数を算出する。保留ビットの増加の処理が無ければ、正規化回数と確定出力ビット数は一致する。

【００７７】

例えば、図４の区間４０１、４０２、４０３はいずれも、その区間の長さが“１２８”以上、“２５６”未満であることを示している。従って、区間４０１、４０２、４０３に対する正規化回数は１回である。区間４０１は、その上端が“５１２”未満であるため、“０”を符号化データのビットとして出力する。区間４０２は、その下端が“５１２”以上であるため、“１”を出力する。

【００７８】

一方、区間４０３は、その下端が“５１２”未満であり、上端が“５１２”以上であるため、保留ビットを１増加させるだけで、この時点での符号化データのビットの出力は行なわない。すなわち、或る区間の上端と下端の間に閾値“５１２”が存在する場合には、保留ビットを“１”だけ増加させ、符号化データの出力は行なわない。

【００７９】

上記を更に詳しく検討すると、次のようになる。

【００８０】

区間の上端、下端を１０ビットで表わすとする。或る区間の上端と下端が共に“５１２”未満であるとき、上端及び下端の最上位ビット（ＭＳＢ）は共に“０”となる。また、或る区間の上端と下端が共に“５１２”以上であるとき、上端及び下端のＭＳＢは共に“１”となる。そして、区間が“５１２”を跨ぐ、すなわち、下端＜５１２≦上端の場合、上端と下端のＭＳＢは一致しない。

【００８１】

従って、上端と下端のＭＳＢが一致するとき、出力ビットの１個が確定する。そして、上端と下端のＭＳＢが不一致の場合、保留ビットを“１”だけ増加させれば良いことになる。複数回の正規化回数の場合も同様に考えることが出来る。

【００８２】

再び、図４に着目する。図４における区間４０４は、その下端が“２７０”、上端が“３８９”、区間の長さが“１１９”（＝３８９－２７０）であることを示している、

先に説明した第１の実施形態の手順によると、２回の正規化回数の結果、２ビットの符号“０１”が出力される。出力ビット数が“２”となるのは、１回目の正規化後の区間の上端と下端が“５１２”を跨がず、さらに上端と下端が２倍された後の２回目の正規化後の区間も“５１２”を跨がないからである。別な言い方をすれば、１回目の正規化後の区間が“２５６”を跨いでいなければ、２倍される２回目の正規化後の区間は“５１２”を跨がない、と言える。すなわち、区間の上端と下端が“５１２”を跨がなければ出力ビット数は１、さらに、その区間が“２５６”も跨がなければ出力ビット数は２、さらに“１２８”も跨がなければ出力ビット数は３というように判定する。結局のところ、区間の上

端と下端を１０ビットで二進表現したとき、両者の最上位ビットからの一致ビット数が出力ビット数に等しいことになる。ただし、出力ビット数は、許容正規化回数を超えることは無いので、許容正規化回数を上限として出力ビット数をクリップする。

【００８３】

図３の説明に戻る。以上のように算出した出力ビット数が決定されるが、ステップＳ３０６では、出力ビット数が“１”以上であるか否かを判定する。出力ビット数が“１”以上であると判定した場合には、ステップＳ３０７乃至ステップＳ３０９の処理を実行する。

【００８４】

このステップＳ３０７乃至Ｓ３０９では、ビット列を符号として出力する。この時、出力されるビット列は、区間の上記の上端と下端の一致ビット列そのものである。

【００８５】

例えば、図４の区間４０４の場合、その上端は“３８９”であり、下端は“２７０”である。これを二進数で表現すると、上端は“０１１０００１０１”、下端は“０１００００１１１０”である。従って、両者は最上位ビットから２ビット“０１”が一致するので、一致ビット列“０１”を出力する。

【００８６】

次にステップＳ３１０では、保留ビット数の更新を行う。前記正規化回数から出力ビット数を減じた値を保留ビット数に加える。この後、ステップＳ３１１において、区間の正規化を行う。下端から前記出力ビット列を除去（ゼロにする）し、前記正規化回数分だけ２倍にする処理を繰り返す。また、区間の長さも同様に前記正規化回数分だけ２倍を繰り返す。ここで２倍をＮ回繰り返すことは、Ｎビットの左シフト算に置き換えても同じである。

【００８７】

上記ステップＳ３０７乃至Ｓ３０９のビット列を出力する工程で、もし保留ビット数が１以上の場合には、まずステップＳ３０７では出力ビット列の先頭ビットのみを出力する。そして、ステップＳ３０８で該先頭ビットの反転ビットを保留ビット数分だけ出力する。そしてステップＳ３０９では、残りの出力ビット列を出力する。

【００８８】

例えば、図４の区間４０４となったときに、保留ビット数が仮に“３”であったとする。この場合、出力ビット列“０１”の先頭ビット“０”をまず出力し、続いて“０”を反転した“１”を保留ビット数分の３個、すなわち“１１１”を出力する。そして、残りのビット列“１”（“０１”の２ビット目）をステップＳ３０９で出力する。よって、このステップで出力される符号化データは“０１１１１”となる。なお、保留ビット数はここでゼロにリセットされる。

【００８９】

以下、ステップＳ３００に戻り、入力シンボルの終了を判定するまで、ステップＳ３０１乃至Ｓ３１１の処理を繰り返し、入力シンボルの終了を判定した場合にはステップＳ３１５の終了処理を行なう。

【００９０】

図３のフローチャートからもわかるように、正規化回数や出力ビットの数を算出して、一括して処理しているため、従来の方法に比べ、ループ構造が格段に少なくなっていることがわかる。また、１入力シンボルに対して１回のループで処理できることもわかる。

【００９１】

図９は、上述した処理を行なうための、本第２の実施形態における算術符号化部１０７の内部ブロック構成図である。本第２の実施形態の算術符号化部１０７は入力部９０１、制御部９０２、終了処理部９０３、出力切替部９０４、区間更新部９０５、正規化部９０６、出力ビット生成部９０７、保留ビット処理部９０８、一致ビット検出部９０９、区間長算出部９１０を備える。以下、各処理部の動作を、図３のフローチャートに対応付けて説明する。

【００９２】
入力部９０１はステップＳ３０１にて、２値化部１０６からの２値シンボルを順次入力する。また、入力部８０１は、入力したシンボル数が、符号化単位である予め設定された個数になったことを検出すると、終了判定信号を制御部９０２に出力する。
【００９３】
区間更新部９０５は、ステップＳ３０２の区間の分割処理と、ステップＳ３０３の区間の選択処理を行なう。そして、区間更新部９０５は、選択された区間の上端と下端の値を示す信号を一致ビット検出部９０９、区間長算出部９１０、正規化部９０６へ供給する。
【００９４】
一致ビット検出部９０９は、区間更新部９０５から供給される上端と下端の値を比較し、２進表現における最上位ビットから連続して一致するビット数をカウントする。この処理は、ステップＳ３０５に相当し、一致したビットの数を制御部９０２へ供給する。
【００９５】
区間長算出部９１０は、区間更新部９０５から供給される上端の値から下端の値を減じて区間の長さを算出する処理（ステップＳ３０４に相当）を行ない、その区間の長さを示す信号を制御部９０２に供給する。
【００９６】
制御部９０２は、区間長算出部９１０より入力した区間の長さから必要な正規化の回数を算出し、正規化回数を正規化部９０６へ供給する。
【００９７】
正規化部９０６は制御部９０２から供給される正規化回数に従い、区間更新部から供給された上端と下端の値を正規化して新たな区間の上端と下端を区間更新部９０５へ供給する。この処理はステップＳ３１１に相当し、ループ処理することなく単一のステップで、前記回数分の正規化を処理する。
【００９８】
また、制御部９０２は各ブロックから供給される信号に基づき、制御信号を各ブロックへ供給する。
【００９９】
制御部９０２は、入力部９０１からの終了判定信号を受信した場合に、終了処理部９０３にトリガ信号を出力する。終了処理部９０３は、制御部９０２からのトリガ信号により、ステップＳ３１５に相当する終了処理を行ない、終了ビット列を出力切替部９０４に出力する。
【０１００】
出力ビット生成部９０７は制御部９０２からの制御信号に従い、“０”もしくは“１”を出力切替部９０４へ供給する。この処理は、図３のフローチャートにおけるステップＳ３０７およびステップＳ３０９に相当し、一致ビット検出部９０９が検出した一致ビット数に相当する個数の出力ビットを生成する。
【０１０１】
保留ビット処理部９０８は制御部９０２からの制御信号に従い、ステップＳ３０８、ステップＳ３１０における保留ビット列を出力切替部９０４へ供給する。
【０１０２】
出力切替部９０４は制御部９０２からの制御信号に従い、終了処理部９０３、出力ビット生成部９０７、保留ビット処理部９０８から供給されるビット列のいずれかを選択し出力する。
【０１０３】
なお、制御部９０２は、図３のフローチャートのステップＳ３００及びＳ３０６に相当する分岐判定処理に相当する処理を行ない、各ブロックの制御信号を発行することになる。
【０１０４】
以上説明したように本第２の実施形態によれば、従来と比べ、算術符号化の処理のルー

プ処理を削減し、また、保留ビット数の増加を抑制することが可能になる。その結果、より少ない回路規模や消費電力で処理可能な算術符号化装置を提供できる。

【０１０５】

また、符号化対象の画像データを入力する入力装置（ビデオカメラ、イメージスキャナやビデオカメラ等の撮像装置、或いは、非圧縮の画像データを蓄積する蓄積装置）を有し、それ以外の図７に示す各構成要素をマイクロコンピュータ等の高度な制御部と、その制御部が実行するプログラムを記憶したＲＡＭによって構成することも勿論可能である。また、図７における算術符号化部１０７のみを、マイクロコンピュータとそれが実行するソフトウェアを格納するメモリによって実現させても構わない。

【０１０６】

また、第１、第２の実施形態では、撮像装置（ビデオカメラ）や映像記録再生装置に適用した例を説明したが、符号化対象情報として２値シンボルを生成する構成を備える装置であれば適用できるので、上記実施形態によって本発明が限定されるものではない。

【０１０７】

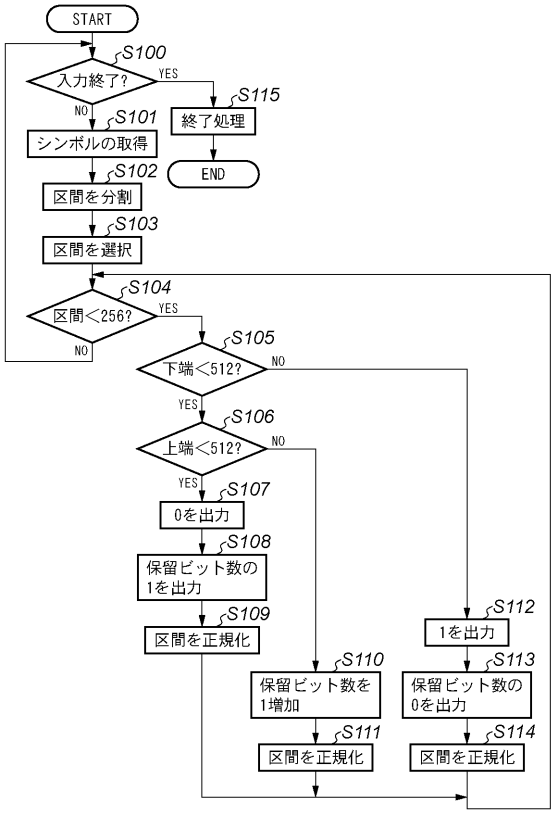
また、通常、ソフトウェア、すなわち、コンピュータプログラムは、ＣＤ－ＲＯＭ等のコンピュータ可読記憶媒体に格納されている。そして、そのコンピュータ可読記憶媒体を、コンピュータの読取り装置（例えばＣＤ－ＲＯＭドライブ）にセットし、システムにコピーもしくはインストールすることで、そのコンピュータが該当するコンピュータプログラムを実行可能にすることができる。従って、かかるコンピュータ可読記憶媒体も本発明の範疇に入るのは明らかである。

【図面の簡単な説明】

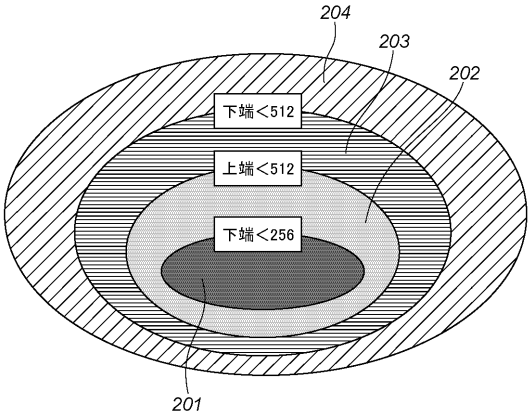
【０１０８】

【図１】第１の実施形態における算術符号化部の処理手順を示すフローチャートである。
【図２】本第１の実施形態における算術符号化処理中の区間の分類を示すベン図である。
【図３】第２の実施形態における算術符号化部の処理手順を示すフローチャートである。
【図４】第２の実施形態における算術符号化における区間の例を示す図である。
【図５】従来の算術符号化処理手順を示すフローチャートである。
【図６】正規化処理を説明するための図である。
【図７】実施形態が適用する装置における符号化部のブロック構成図である。
【図８】第１の実施形態における算術符号化部の具体的なブロック構成図である。
【図９】第２の実施形態における算術符号化部の具体的なブロック構成図である。

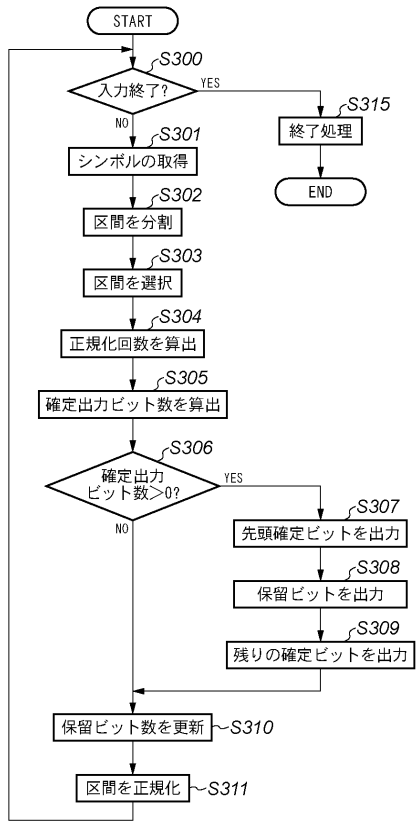
【図１】


START
S100
入力終了?
YES
NO
S115
終了処理
END
S101
シンボルの取得
S102
区間を分割
S103
区間を選択
S104
区間＜256?
YES
NO
S105
下端＜512?
NO
YES
S106
上端＜512?
NO
YES
S107
0を出力
S108
保留ビット数の
1を出力
S109
区間を正規化
S110
保留ビット数を
1増加
S111
区間を正規化
S112
1を出力
S113
保留ビット数の
0を出力
S114
区間を正規化

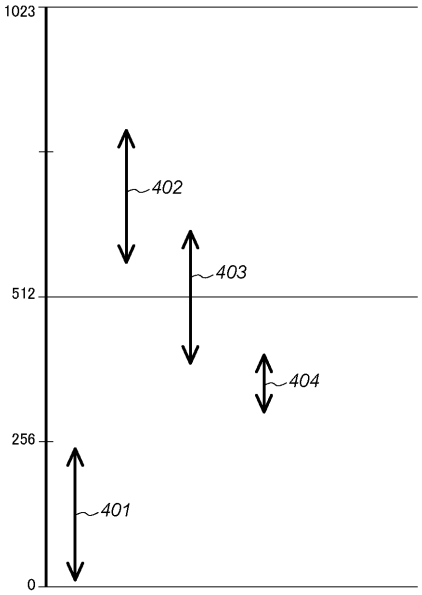
【図２】


204
203
202
下端＜512
上端＜512
下端＜256
201

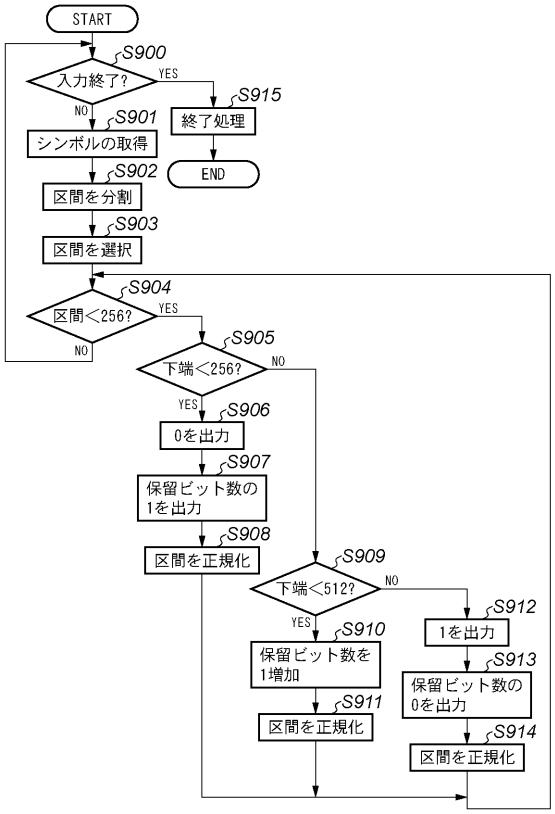
【図３】


START
S300
入力終了?
YES
NO
S315
終了処理
END
S301
シンボルの取得
S302
区間を分割
S303
区間を選択
S304
正規化回数を算出
S305
確定出力ビット数を算出
S306
確定出力
ビット数＞0?
YES
NO
S307
先頭確定ビットを出力
S308
保留ビットを出力
S309
残りの確定ビットを出力
保留ビット数を更新
S310
区間を正規化
S311

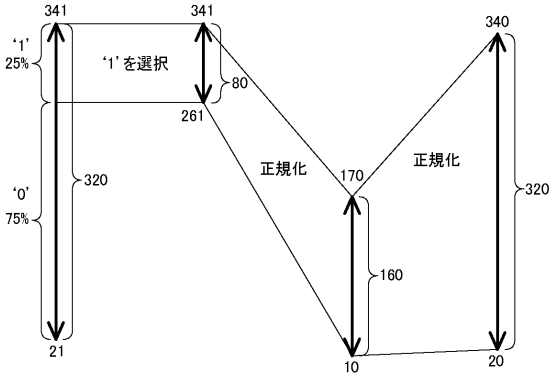
【図４】


1023
512
256
0
402
403
404
401

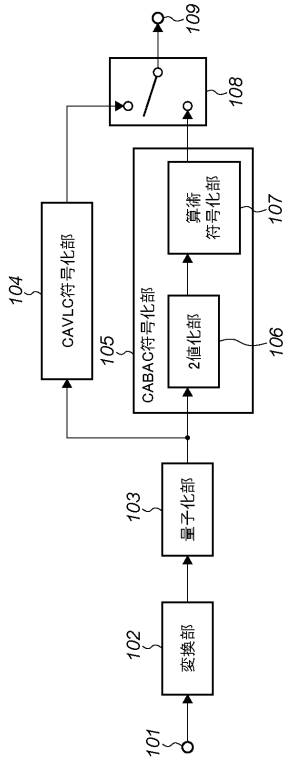
【図５】


START
S900
入力終了?
YES
NO
S915
終了処理
END
S901
シンボルの取得
S902
区間を分割
S903
区間を選択
S904
区間＜256?
YES
NO
S905
下端＜256?
NO
YES
S906
0を出力
S907
保留ビット数の
1を出力
S908
区間を正規化
S909
下端＜512?
NO
YES
S910
保留ビット数を
1増加
S911
区間を正規化
S912
1を出力
S913
保留ビット数の
0を出力
S914
区間を正規化

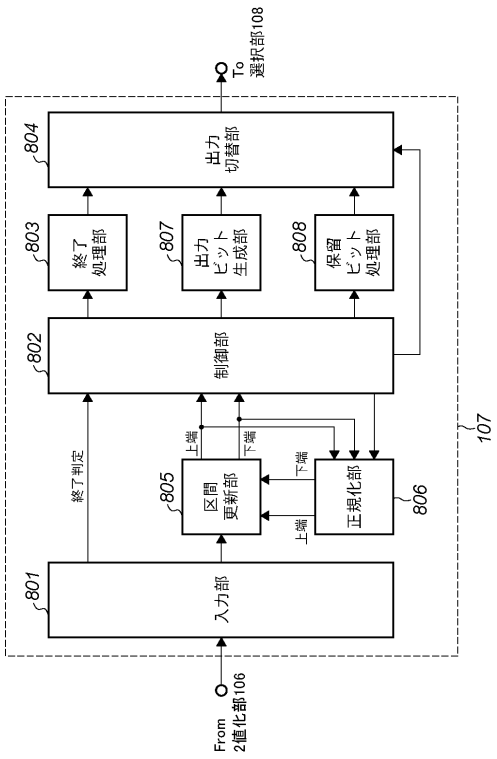
【図６】


341
341
340
'1'
25%
'1'を選択
80
261
正規化
170
正規化
'0'
75%
320
320
160
21
10
20

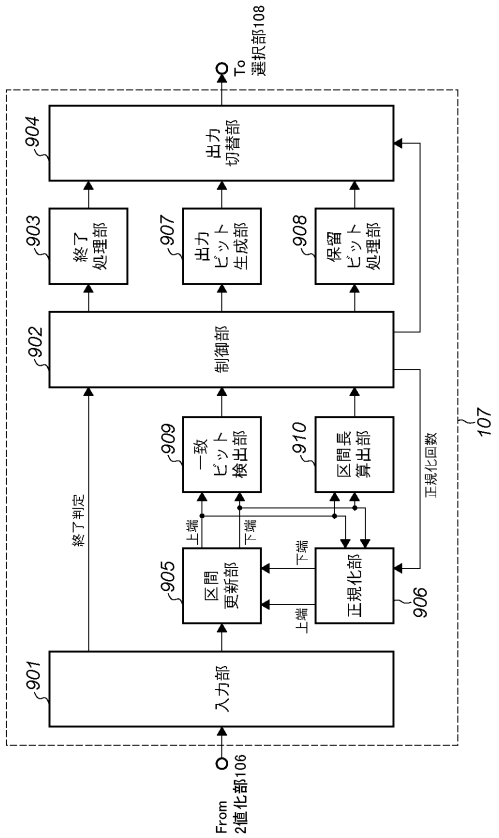
【図７】


109
108
算術
符号化部
107
CAVLC符号化部
104
CABAC符号化部
105
2値化部
106
量子化部
103
変換部
102
101

【図８】


To
選択部108
804
出力
切替部
803
終了
処理部
807
出力
ビット
生成部
808
保留
ビット
処理部
802
制御部
上端
下端
終了判定
805
区間
更新部
下端
上端
正規化部
806
107
801
入力部
From
2値化部106

【図９】


To
選択部108
904
出力
切替部
903
終了
処理部
907
出力
ビット
生成部
908
保留
ビット
処理部
902
制御部
909
一致
ビット
検出部
910
区間長
算出部
107
終了判定
上端
下端
905
区間
更新部
下端
上端
正規化部
906
正規化回数
901
入力部
From
2値化部106

フロントページの続き

審査官　北村　智彦

(56)参考文献　特開２００４－１３５２５１（ＪＰ，Ａ）
特開２００５－２５２３７４（ＪＰ，Ａ）
特表２００５－５２５０１８（ＪＰ，Ａ）
国際公開第２００８／００１０１２（WO，Ａ１）
特開平１０－０３２４９６（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０３Ｍ３／００－１１／００
Ｈ０４Ｎ　　７／３０