

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4547503号
(P4547503)

(45) 発行日 平成22年9月22日(2010.9.22)

(24) 登録日 平成22年7月16日(2010.7.16)

(51) Int.Cl. F I
HO 3M 7/40 (2006.01) HO 3M 7/40
HO 4N 7/26 (2006.01) HO 4N 7/13 Z

請求項の数 9 (全 35 頁)

| | | | |
|---------------|------------------------------|-----------|--|
| (21) 出願番号 | 特願2008-503883 (P2008-503883) | (73) 特許権者 | 304020292 |
| (86) (22) 出願日 | 平成19年3月6日(2007.3.6) | | 国立大学法人徳島大学 |
| (86) 国際出願番号 | PCT/JP2007/054358 | | 徳島県徳島市新蔵町2丁目24番地 |
| (87) 国際公開番号 | W02007/102518 | (74) 代理人 | 100104949 |
| (87) 国際公開日 | 平成19年9月13日(2007.9.13) | | 弁理士 豊栖 康司 |
| 審査請求日 | 平成20年7月4日(2008.7.4) | (74) 代理人 | 100074354 |
| (31) 優先権主張番号 | 特願2006-61780 (P2006-61780) | | 弁理士 豊栖 康弘 |
| (32) 優先日 | 平成18年3月7日(2006.3.7) | (72) 発明者 | 宋 天 |
| (33) 優先権主張国 | 日本国(JP) | | 徳島県徳島市南常三島町2丁目1番地 国立大学法人徳島大学大学院ソシオテクノサイエンス研究部内 |
| | | (72) 発明者 | 山田 篤 |
| | | | 徳島県徳島市南常三島町2丁目1番地 国立大学法人徳島大学大学院工学研究科内 |

最終頁に続く

(54) 【発明の名称】 算術符号化装置、算術符号化方法、算術符号化プログラム及びプログラムを格納したコンピュータで読み取り可能な記録媒体

(57) 【特許請求の範囲】

【請求項1】

多値信号を2値化するための2値化部と、
 前記2値化部で生成された2値シンボルを算術符号化するための算術符号化部と、
 コンテキスト番号を生成するためのコンテキスト計算部と、
 前記2値化部で生成された2値シンボルに基づいて、優勢シンボル或いは劣勢シンボルが示すシンボル値とシンボル出現確率とを更新するための更新部と、
 前記コンテキスト計算部で計算されたコンテキスト番号をインデックスとして、前記シンボル値と前記シンボル出現確率とを格納するための確率状態格納部と、
 を備える算術符号化装置であって、

前記算術符号化部は、

劣勢シンボルの区間を発生確率に基づいて分割した領域 r L P S の予測値を格納する第1格納部と、

同じく r L P S の予測値を格納する、前記第1格納部とは別個の第2格納部と、

前記第1格納部及び第2格納部に格納された r L P S の予測値を参照して算術符号化を行う演算部と、

を備えてなることを特徴とする算術符号化装置。

【請求項2】

多値信号を2値化するための2値化部と、

前記2値化部で生成された2値シンボルを算術符号化するための算術符号化部と、

コンテキスト番号を生成するためのコンテキスト計算部と、
前記 2 値化部で生成された 2 値シンボルに基づいて、優勢シンボル或いは劣勢シンボル
が示すシンボル値とシンボル出現確率とを更新するための更新部と、

前記コンテキスト計算部で計算されたコンテキスト番号をインデックスとして、前記シン
ボル値と前記シンボル出現確率とを格納するための確率状態格納部と、
を備える算術符号化装置であって、

前記算術符号化部は、

優勢シンボル処理時の優勢シンボル処理連続数、初期の区間、シンボル出現確率に基づ
いて決定される更新区間の予測値を格納する更新区間格納部と、

優勢シンボル処理時の優勢シンボル処理連続数、及び初期のシンボル出現確率に基づ
いて決定されるシンボル出現確率の予測値を格納するためのコンテキスト更新テーブルと
、

再正規化の回数に基づいて、更新区間の補集合である low を更新する再正規化テー
ブルを有する再正規化部と、

を備えることを特徴とする算術符号化装置。

【請求項 3】

多値信号を 2 値化するための 2 値化部と、

前記 2 値化部で生成された 2 値シンボルを算術符号化するための算術符号化部と、

コンテキスト番号を生成するためのコンテキスト計算部と、

前記 2 値化部で生成された 2 値シンボルに基づいて、優勢シンボル或いは劣勢シンボル
が示すシンボル値とシンボル出現確率とを更新するための更新部と、

前記コンテキスト計算部で計算されたコンテキスト番号をインデックスとして、前記シン
ボル値と前記シンボル出現確率とを格納するための確率状態格納部と、

を備える算術符号化装置であって、

前記算術符号化部は、

優勢シンボル処理時の優勢シンボル処理連続数、初期の区間、シンボル出現確率に基づ
いて決定される更新区間の予測値を格納する第 1 更新区間格納部と、

同じく更新区間の予測値を格納する、前記第 1 更新区間格納部と別個の第 2 更新区間
格納部と、

優勢シンボル処理時の優勢シンボル処理連続数、及び初期のシンボル出現確率に基づ
いて決定されるシンボル出現確率の予測値を格納するための第 1 コンテキスト更新テー
ブルと、

優勢シンボル処理時の優勢シンボル処理連続数、及び前記第 1 コンテキスト更新テー
ブルで更新されたシンボル出現確率に基づいて決定されるシンボル出現確率の予測値を格
納するための、前記第 1 コンテキスト更新テーブルと別個の第 2 コンテキスト更新テー
ブルと、

再正規化の回数に基づいて、更新区間の補集合である low を更新する再正規化テー
ブルを有する再正規化部と、

を備えることを特徴とする算術符号化装置。

【請求項 4】

多値信号を 2 値化するための 2 値化部と、

前記 2 値化部で生成された 2 値シンボルを算術符号化するための算術符号化部と、

コンテキスト番号を生成するためのコンテキスト計算部と、

前記 2 値化部で生成された 2 値シンボルに基づいて、優勢シンボル或いは劣勢シンボル
が示すシンボル値とシンボル出現確率とを更新するための更新部と、

前記コンテキスト計算部で計算されたコンテキスト番号をインデックスとして、前記シン
ボル値と前記シンボル出現確率とを格納するための確率状態格納部と、

を備える算術符号化装置であって、

前記算術符号化部が、

劣勢シンボルの区間を発生確率に基づいて分割した領域 r L P S の予測値を格納する

10

20

30

40

50

第1格納部と、

同じく r L P S の予測値を格納する、前記第1格納部とは別個の第2格納部と、

前記第1格納部及び第2格納部に格納された r L P S の予測値を参照して算術符号化を行う演算部と、

を備える2サイクル予測部と、

優勢シンボル処理時の優勢シンボル処理連続数、初期の区間、シンボル出現確率に基づいて決定される更新区間の予測値を格納する更新区間格納部と、

優勢シンボル処理時の優勢シンボル処理連続数、及び初期のシンボル出現確率に基づいて決定されるシンボル出現確率の予測値を格納するためのコンテキスト更新テーブルと、

再正規化の回数に基づいて、更新区間の補集合である low を更新する再正規化テーブルを有する再正規化部と、

を備える連続優勢シンボル処理予測部と

前記2サイクル予測部と連続優勢シンボル処理予測部のいずれかを切り替える予測手法切替部と、

を備えることを特徴とする算術符号化装置。

【請求項5】

2値シンボルの算術符号化にあたり、出現確率の高いシンボルである優勢シンボルと出現確率の低いシンボルである劣勢シンボルとを、前シンボルまでの数線上の区間及び劣勢シンボルの数線上の区間の座標値に基づいて、数線上の所定の区間に対応させ、

その数線上の区間対応結果と入力される2値シンボルから符号化信号を出力する算術符号化方法であって、

符号化対象のシンボルが優勢シンボルである場合の優勢シンボル処理時の区間座標値、及び劣勢シンボルである場合の劣勢シンボル処理時の区間座標値を予め記録した第1 r L P S テーブルを参照して、第1シンボルの区間を更新した分割領域を求める工程と、

得られた分割領域と、第1 r L P S テーブルと別個の第2 r L P S テーブルを参照して、第1シンボルに続く第2シンボルの区間を更新した分割領域を求める工程と、

必要に応じて再正規化を行い、出力符号を出力する工程と、

コンテキスト番号を更新する工程と、

を含むことを特徴とする算術符号化方法。

【請求項6】

2値シンボルの算術符号化にあたり、出現確率の高いシンボルである優勢シンボルと出現確率の低いシンボルである劣勢シンボルとを、前シンボルまでの数線上の区間及び劣勢シンボルの数線上の区間の座標値に基づいて、数線上の所定の区間に対応させ、

その数線上の区間対応結果と入力される2値シンボルから符号化信号を出力する算術符号化方法であって、

2値シンボルの優勢シンボルが連続する場合に、優勢シンボルの連続数とコンテキスト番号に基づいて、更新区間格納部を参照して更新区間の予測値を求める工程と、

コンテキスト番号を格納するコンテキスト更新テーブルを用いて更新する工程と、

を含むことを特徴とする算術符号化方法。

【請求項7】

2値シンボルの算術符号化にあたり、出現確率の高いシンボルである優勢シンボルと出現確率の低いシンボルである劣勢シンボルとを、前シンボルまでの数線上の区間及び劣勢シンボルの数線上の区間の座標値に基づいて、数線上の所定の区間に対応させ、

その数線上の区間対応結果と入力される2値シンボルから符号化信号を出力する算術符号プログラムであって、

符号化対象のシンボルが優勢シンボルである場合の優勢シンボル処理時の区間座標値、及び劣勢シンボルである場合の劣勢シンボル処理時の区間座標値を予め記録した第1 r L P S テーブルを参照して、第1シンボルの区間を更新した分割領域を求める機能と、

得られた分割領域と、第1 r L P S テーブルと別個の第2 r L P S テーブルを参照して

10

20

30

40

50

、第1シンボルに続く第2シンボルの区間を更新した分割領域を求める機能と、
必要に応じて再正規化を行い、出力符号を出力する機能と、
コンテキスト番号を更新する機能と、
をコンピュータに実現させることを特徴とする算術符号化プログラム。

【請求項8】

2値シンボルの算術符号化にあたり、出現確率の高いシンボルである優勢シンボルと出現確率の低いシンボルである劣勢シンボルとを、前シンボルまでの数線上の区間及び劣勢シンボルの数線上の区間の座標値に基づいて、数線上の所定の区間に対応させ、
その数線上の区間対応結果と入力される2値シンボルから符号化信号を出力する算術符号化プログラムであって、

10

2値シンボルの優勢シンボルが連続する場合に、優勢シンボルの連続数とコンテキスト番号に基づいて、更新区間格納部を参照して更新区間の予測値を求める機能と、
コンテキスト番号を格納するコンテキスト更新テーブルを用いて更新する機能と、
をコンピュータに実現させることを特徴とする算術符号化プログラム。

【請求項9】

請求項7又は8に記載されるプログラムを格納したコンピュータで読み取り可能な記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

20

本発明は、動画像符号化・復号に用いられる算術符号化技術に関するものであり、特に、動画像符号化規格であるH.264/AVCにおけるCABAC (Context-based Adaptive Binary Arithmetic Coding: コンテキスト準拠適応2進算術符号化) に準拠した算術符号化装置、算術符号化方法、算術符号化プログラム及びプログラムを格納したコンピュータで読み取り可能な記録媒体に関する。

【背景技術】

【0002】

近年、DVDや地上デジタルハイビジョン等の高画質動画像の普及が急速に進んでおり、動画像の高ビットレート、高解像度によって情報量が飛躍的に増加し、効率のよい動画像データの圧縮が必須となっている。デジタル動画像圧縮の主な構成技術は、予測符号化、動き処理、変換符号化、量子化、符号割当の5つがある。この内、符号割当は、量子化レベルなどに「0」、「1」の符号を効率よく割り当て、冗長な符号の削減をする。符号割当には可変長符号化、適応符号化、固定長符号化等が知られている。可変長符号化 (VLC: Variable Length Coding) はハフマン符号化、ランレングス符号化、算術符号化、適応ビット割当等に区分される。この内、算術符号化は、シンボルの出現確率に応じて符号化テーブルを作成し符号割当を行うものである。

30

【0003】

算術符号化は、画像データの特徴に応じた最適な符号割当ができるため、固定した符号化テーブルを使用するハフマン符号化などに比べて符号化効率が優れている。しかしながら算術符号化は、シンボルの出現確率を計算しながら符号化テーブルを作成し符号割当をする動的な方式のため、一般的にハフマン符号化と比較して多大な計算処理を必要とするという問題がある。

40

【0004】

一方、ITU-T (International Telecommunication Union - Telecommunication Standardization Sector) では、動画像符号化・復号技術としてH.264の規格策定を行っている (非特許文献1参照)。特許文献1には、ITU-T規格H.264による算術符号化・復号の処理プロセスの具体的な実現方法の実例が開示される。図26は、H.264/AVCによる算術符号化・復号の処理プロセスのフローチャートである。同図は、非特許文献1のFig. 9-1を、機能を中心に再編集したものである。図26を用いて、算術符号化・復号の流れを以下に説明する。

50

【 0 0 0 5 】

あるシンタックス要素 (Syntax Element : S E) について、符号化又は復号が開始されると、ステップ 1 において、そのシンタックス要素に対して、コンテキスト計算が行われる。コンテキスト計算では、コンテキストに対応するコンテキスト番号が求められ、コンテキスト番号によりシンボル値と確率状態が求められる。シンボル値は、一部の例外はあるものの、多くは対応表を参照することによって求められる。確率状態は、後述のステップ 3 で格納されている値を参照して求められる。求められたシンボル値と確率状態は、算術符号化・復号プロセスに与えられ、ステップ 2 に進む。

【 0 0 0 6 】

ステップ 2 においては、ステップ 1 より受け取ったシンボル値と確率状態を用いて、当該シンタックスの算術符号化又は復号を行い、結果を出力信号として出力する。同時に、符号化又は復号したシンボルの情報を、シンボル出現確率制御プロセスに出力し、ステップ 3 に進む。

10

【 0 0 0 7 】

ステップ 3 においては、符号化または復号したシンボルの情報を受けて、次の処理で用いるシンボル値と確率状態の更新値を求め、確率状態として格納しておく。

【 0 0 0 8 】

ステップ 4 において、当該シンタックス要素の符号化又は復号が完了したかどうかを判定し、完了してない場合には、ステップ 1 に戻り、符号化又は復号が完了するまで同様のプロセスを繰り返す。当該シンタックス要素の符号化又は復号が完了すると、次のシンタックス要素について符号化又は復号を行う。

20

【 0 0 0 9 】

図 2 7 に、H . 2 6 4 / A V C の算術符号化の処理プロセスを具現化した算術符号化装置のブロック図を示す。この算術符号化装置 7 0 0 は、適応算術符号化・復号部 7 1 0、コンテキスト計算部 7 2 0、及び符号化・復号制御部 7 3 0 を備え、適応算術符号化・復号部 7 1 0 は、算術符号化・復号部 7 1 1、シンボル出現確率制御部 7 1 2、及び確率状態格納部 7 1 3 を有する。コンテキスト計算部 7 2 0 は、入力信号 (S i n) S 1 1 のシンタックス要素 (S E) の種別と既符号化ビット数又は復号ビット数より、コンテキスト番号を生成する (図 2 6 のステップ 1 の処理に相当する) 。

【 0 0 1 0 】

この算術符号化装置 7 0 0 では、各処理ステップは、その前のステップでの処理結果を受けて、そのステップでの処理を開始する必要があるため、その動作は、シーケンシャルな処理となり、一つのシンタックス要素が処理される時間は、各処理ステップの和となる。すなわち、例えば、シンタックス要素 0 の処理に必要な区間 S 0 は、処理 5 1 と処理 5 2 と処理 5 3 と処理 5 4 とのそれぞれに費やされる時間の和となる。

30

【 0 0 1 1 】

符号化又は復号の時間当たりの処理符号量 (即ちシンタックス要素数) が多い場合、符号化又は復号の処理時間を短縮することは、必須の要件となる。本検討例の算術符号化・復号装置 4 0 では、各ステップは、シーケンシャルな処理を行っているため、全体の処理時間の短縮のためには、各ステップの処理時間を短縮しなければならず、現状の技術レベルでは、高速化には限界がある。

40

【 0 0 1 2 】

これに対して、特許文献 1 には、算術符号化処理を高速化した算術符号化装置が開示される。この算術符号化装置 8 0 0 は、図 2 8 のブロック図に示すように適応算術符号化部 8 1 0、コンテキスト計算部 8 2 0、及び符号化制御部 8 3 0 を備える。コンテキスト計算部 8 2 0 は、レジスタ 8 2 1 と比較判定部 8 2 2 とを有する。適応算術符号化部 8 1 0 は、算術符号化部 8 1 1 とシンボル出現確率制御部 8 1 4 と確率状態格納部 8 1 5 とを有する。これによれば、ある入力信号が算術符号化部 8 1 1 で符号化される間に、コンテキスト計算部 8 2 0 は、次の入力信号についてコンテキストを求めることができるので、算術符号化部 8 1 1 は、ある入力信号を符号化後、直ちに次の入力信号の符号化処理を行え

50

る。従って、符号化処理の高速化が可能となる。

【 0 0 1 3 】

さらに、ある信号が算術符号化部 8 1 1 で符号化される間に、シンボル出現確率制御部 8 1 4 は、算術符号化部 8 1 1 による符号化シンボルが、優勢シンボルである場合と劣勢シンボルである場合とについて、それぞれのシンボル値と確率状態の更新値を求めておき、符号化シンボルが確定した後に、一方の更新値を選択する。この結果、算術符号化部 8 1 1 は、次の符号の符号化において、必要なシンボル値と確率状態を得て、直ちに符号化処理と確率状態記録部 8 1 5 への書込み処理を行えるため、算術符号化処理の更なる高速化が可能となる。

【特許文献 1】特開 2 0 0 5 - 1 3 0 0 9 9 号公報

10

【非特許文献 1】Joint Video Team(JVT) of ISO/IEC MPEG & ITU-TVCEG; "Draft I TU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264|ISO/IEC 14496-10 MPEG-AVC)."

【非特許文献 2】Hassan Shojania and Subramania Sudharsanan, "A VLSI Architecture for High Performance CABAC Encoding." Visual Communications and Image Processing 2005, Proc. of SPIE Vol. 5960, pp.1444.

【非特許文献 3】大久保榮他「改訂版 H . 2 6 4 / A V C 教科書」インプレス

【非特許文献 4】「技術分野別特許マップ 電気 1 4 デジタル動画像圧縮技術」特許庁

【非特許文献 5】Roberto R. Osorio, Javier D. Bruguera, "High-Throughput Architecture for H.264/AVC CABAC Compression System" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 16, NO. 11, NOVEMBER 2006.

20

【発明の開示】

【発明が解決しようとする課題】

【 0 0 1 4 】

しかしながらこの方法でも、1 サイクルで 1 ビットの符号化を実現するものでなく、高速化対策としては十分ではない。一方、非特許文献 2 では 1 サイクルで 1 バイナリシンボルを処理可能な高速化手法が提案されているが、この手法でも十分でなく 1 サイクルで複数の処理を行うには至っておらず、十分な高速処理が実現できるとは言い難い。

【 0 0 1 5 】

H . 2 6 4 / A V C の符号化処理処理においては、エントロピー符号化である可変長符号化 (V L C) が採用されている。V L C は、実際に H . 2 6 4 / A V C のビットストリームを生成しているモジュールである。この符号化方式として C A B A C と C A V L C がある。C A V L C (Context-Adaptive Variable Length Coding : コンテキスト適応型可変長符号化方式) は、量子化された D C T 係数値の符号化にのみ用いられる。一方 C A B A C (Context Adaptive Binary Arithmetic Coding : コンテキスト適応型 2 値算術符号化方式) は、算術符号化を用いたエントロピー符号化方式であり、C A V L C と比較して符号化効率が高いという利点を有する。

30

【 0 0 1 6 】

しかしながら、C A B A C 処理は高い符号化効率を得ることができる反面、処理時間が長いという問題がある。特に高解像度アプリケーションにおいて、リアルタイムに符号化を行うには、処理に極めて高い動作周波数を要する。既存の M P U や C P U では、計算上数 G H z もの動作周波数が要求されることとなって、ハイビジョン放送などのメディアに対応できないという問題があった。このため、C A B A C 処理を高速に行うための構成が切望されているものの、上述の通り従来の手法では未だ不十分であった。

40

【 0 0 1 7 】

本発明は、このような問題点を解消するためになされたものである。本発明の主な目的は、H . 2 6 4 / A V C における C A B A C の処理の演算量を削減し、実装を容易にした算術符号化装置、算術符号化方法、算術符号化プログラム及びプログラムを格納したコンピュータで読み取り可能な記録媒体を提供することにある。

【課題を解決するための手段】

50

【0018】

上記の目的を達成するために、本発明の第1の算術符号化装置は、多値信号を2値化するための2値化部と、2値化部で生成された2値シンボルを算術符号化するための算術符号化部と、コンテキスト番号を生成するためのコンテキスト計算部と、2値化部で生成された2値シンボルに基づいて、優勢シンボル或いは劣勢シンボルが示すシンボル値とシンボル出現確率とを更新するための更新部と、コンテキスト計算部で計算されたコンテキスト番号をインデックスとして、シンボル値とシンボル出現確率とを格納するための確率状態格納部とを備える算術符号化装置であって、算術符号化部は、劣勢シンボルの区間を発生確率に基づいて分割した領域 $r L P S$ の予測値を格納する第1格納部と、同じく $r L P S$ の予測値を格納する、第1格納部とは別個の第2格納部と、第1格納部及び第2格納部に格納された $r L P S$ の予測値を参照して算術符号化を行う演算部とを備える。これにより、1サイクル中に2ビット以上の処理が可能な高速な算術符号化を実現できる。

10

【0019】

さらに第2の算術符号化装置は、多値信号を2値化するための2値化部と、2値化部で生成された2値シンボルを算術符号化するための算術符号化部と、コンテキスト番号を生成するためのコンテキスト計算部と、2値化部で生成された2値シンボルに基づいて、優勢シンボル或いは劣勢シンボルが示すシンボル値とシンボル出現確率とを更新するための更新部と、コンテキスト計算部で計算されたコンテキスト番号をインデックスとして、シンボル値とシンボル出現確率とを格納するための確率状態格納部とを備える算術符号化装置であって、算術符号化部は、優勢シンボル処理時の優勢シンボル処理連続数、初期の区間、シンボル出現確率に基づいて決定される更新区間の予測値を格納する更新区間格納部と、優勢シンボル処理時の優勢シンボル処理連続数、及び初期のシンボル出現確率に基づいて決定されるシンボル出現確率の予測値を格納するためのコンテキスト更新テーブルと、再正規化の回数に基づいて、更新区間の補集合である low を更新する再正規化テーブルを有する再正規化部とを備える。これにより、連続して優勢シンボル処理が行われる場合の算術符号化処理を数サイクル削減でき、処理の低減と高速化が図られる。

20

【0020】

さらにまた第3の算術符号化装置は、多値信号を2値化するための2値化部と、2値化部で生成された2値シンボルを算術符号化するための算術符号化部と、コンテキスト番号を生成するためのコンテキスト計算部と、2値化部で生成された2値シンボルに基づいて、優勢シンボル或いは劣勢シンボルが示すシンボル値とシンボル出現確率とを更新するための更新部と、コンテキスト計算部で計算されたコンテキスト番号をインデックスとして、シンボル値とシンボル出現確率とを格納するための確率状態格納部とを備える算術符号化装置であって、算術符号化部は、優勢シンボル処理時の優勢シンボル処理連続数、初期の区間、シンボル出現確率に基づいて決定される更新区間の予測値を格納する第1更新区間格納部と、同じく更新区間の予測値を格納する、第1更新区間格納部と別個の第2更新区間格納部と、優勢シンボル処理時の優勢シンボル処理連続数、及び初期のシンボル出現確率に基づいて決定されるシンボル出現確率の予測値を格納するための第1コンテキスト更新テーブルと、優勢シンボル処理時の優勢シンボル処理連続数、及び前記第1コンテキスト更新テーブルが更新したシンボル出現確率に基づいて決定されるシンボル出現確率の予測値を格納するための、第1コンテキスト更新テーブルと別個の第2コンテキスト更新テーブルと、再正規化の回数に基づいて、更新区間の補集合である low を更新する再正規化テーブルを有する再正規化部とを備える。これにより、連続して優勢シンボル処理が行われる場合の算術符号化処理を、複数の異なるコンテキストが使用される場合でも削減できる。

30

40

【0021】

さらにまた第4の算術符号化装置は、多値信号を2値化するための2値化部と、2値化部で生成された2値シンボルを算術符号化するための算術符号化部と、コンテキスト番号を生成するためのコンテキスト計算部と、2値化部で生成された2値シンボルに基づいて、優勢シンボル或いは劣勢シンボルが示すシンボル値とシンボル出現確率とを更新するた

50

めの更新部と、コンテキスト計算部で計算されたコンテキスト番号をインデックスとして、シンボル値とシンボル出現確率とを格納するための確率状態格納部とを備える算術符号化装置であって、算術符号化部が、劣勢シンボルの区間を発生確率に基づいて分割した領域 $r L P S$ の予測値を格納する第 1 格納部と、同じく $r L P S$ の予測値を格納する、第 1 格納部とは別個の第 2 格納部と、第 1 格納部及び第 2 格納部に格納された $r L P S$ の予測値を参照して算術符号化を行う演算部とを備える 2 サイクル予測部と、優勢シンボル処理時の優勢シンボル処理連続数、初期の区間、シンボル出現確率に基づいて決定される更新区間の予測値を格納する更新区間格納部と、優勢シンボル処理時の優勢シンボル処理連続数、及び初期のシンボル出現確率に基づいて決定されるシンボル出現確率の予測値を格納するためのコンテキスト更新テーブルと、再正規化の回数に基づいて、更新区間の補集合である low を更新する再正規化テーブルを有する再正規化部とを備える連続優勢シンボル処理予測部と、2 サイクル予測部と連続優勢シンボル処理予測部のいずれかを切り替える予測手法切替部とを備える。これにより、2 サイクル予測又は連続優勢シンボル処理予測を適宜切り替えて、サイクル数を効率よく削減して更に高速な算術符号化が実現できる。

10

【 0 0 2 2 】

一方、算術符号化方法は、2 値シンボルの算術符号化にあたり、出現確率の高いシンボルである優勢シンボルと出現確率の低いシンボルである劣勢シンボルとを、前シンボルまでの数線上の区間及び劣勢シンボルの数線上の区間の座標値に基づいて、数線上の所定の区間に対応させ、その数線上の区間対応結果と入力される 2 値シンボルから符号化信号を出力する算術符号化方法であって、符号化対象のシンボルが優勢シンボルである場合の優勢シンボル処理時の区間座標値、及び劣勢シンボルである場合の劣勢シンボル処理時の区間座標値を予め記録した第 1 $r L P S$ テーブルを参照して、第 1 シンボルの区間を更新した分割領域を求める工程と、得られた分割領域と、第 1 $r L P S$ テーブルと別個の第 2 $r L P S$ テーブルを参照して、第 1 シンボルに続く第 2 シンボルの区間を更新した分割領域を求める工程と、必要に応じて再正規化を行い、出力符号を出力する工程と、コンテキスト番号を更新する工程とを含む。これにより、1 サイクル中に 2 ビット以上の処理が可能な高速な算術符号化を実現できる。

20

【 0 0 2 3 】

また算術符号化方法は、2 値シンボルの算術符号化にあたり、出現確率の高いシンボルである優勢シンボルと出現確率の低いシンボルである劣勢シンボルとを、前シンボルまでの数線上の区間及び劣勢シンボルの数線上の区間の座標値に基づいて、数線上の所定の区間に対応させ、その数線上の区間対応結果と入力される 2 値シンボルから符号化信号を出力する算術符号化方法であって、2 値シンボルの優勢シンボルが連続する場合に、優勢シンボルの連続数とコンテキスト番号に基づいて、更新区間格納部を参照して更新区間の予測値を求める工程と、コンテキスト番号を格納するコンテキスト更新テーブルを用いて更新する工程とを含む。これにより、連続して優勢シンボル処理が行われる場合の算術符号化処理を数サイクル削減でき、処理の低減と高速化が図られる。

30

【 0 0 2 4 】

さらに、算術符号化プログラムは、2 値シンボルの算術符号化にあたり、出現確率の高いシンボルである優勢シンボルと出現確率の低いシンボルである劣勢シンボルとを、前シンボルまでの数線上の区間及び劣勢シンボルの数線上の区間の座標値に基づいて、数線上の所定の区間に対応させ、その数線上の区間対応結果と入力される 2 値シンボルから符号化信号を出力する算術符号プログラムであって、符号化対象のシンボルが優勢シンボルである場合の優勢シンボル処理時の区間座標値、及び劣勢シンボルである場合の劣勢シンボル処理時の区間座標値を予め記録した第 1 $r L P S$ テーブルを参照して、第 1 シンボルの区間を更新した分割領域を求める機能と、得られた分割領域と、第 1 $r L P S$ テーブルと別個の第 2 $r L P S$ テーブルを参照して、第 1 シンボルに続く第 2 シンボルの区間を更新した分割領域を求める機能と、必要に応じて再正規化を行い、出力符号を出力する機能と、コンテキスト番号を更新する機能とをコンピュータに実現させる。これにより、1 サイ

40

50

クル中に2ビット以上の処理が可能な高速な算術符号化を実現できる。

【0025】

さらにまた算術符号化プログラムは、2値シンボルの算術符号化にあたり、出現確率の高いシンボルである優勢シンボルと出現確率の低いシンボルである劣勢シンボルとを、前シンボルまでの数線上の区間及び劣勢シンボルの数線上の区間の座標値に基づいて、数線上の所定の区間に対応させ、その数線上の区間対応結果と入力される2値シンボルから符号化信号を出力する算術符号化プログラムであって、2値シンボルの優勢シンボルが連続する場合に、優勢シンボルの連続数とコンテキスト番号に基づいて、更新区間格納部を参照して更新区間の予測値を求める機能と、コンテキスト番号を格納するコンテキスト更新テーブルを用いて更新する機能とをコンピュータに実現させる。これにより、連続して優勢シンボル処理が行われる場合の算術符号化処理を数サイクル削減でき、処理の低減と高速化が図られる。

10

【0026】

またコンピュータで読み取り可能な記録媒体は、上記プログラムを格納するものである。記録媒体には、CD-ROM、CD-R、CD-RWやフレキシブルディスク、磁気テープ、MO、DVD-ROM、DVD-RAM、DVD-R、DVD+R、DVD-RW、DVD+RW、Blue-ray（登録商標）、HD DVD（登録商標）（AOD）等の磁気ディスク、光ディスク、光磁気ディスク、半導体メモリその他のプログラムを格納可能な媒体が含まれる。またプログラムには、上記記録媒体に格納されて配布されるものの他、インターネット等のネットワーク回線を通じてダウンロードによって配布される形態のものも含まれる。さらに記録媒体には、上記プログラムがソフトウェアやファームウェア等の形態で実行可能な状態に実装された汎用もしくは専用機器を含む。さらにまたプログラムに含まれる各処理や機能は、コンピュータで実行可能なプログラムソフトウェアにより実行してもよいし、各部の処理を所定のゲートアレイ（FPGA、ASIC）等のハードウェア、又はプログラム・ソフトウェアとハードウェアの一部の要素を実現する部分的ハードウェア・モジュールとが混在する形式で実現してもよい。

20

【発明の効果】

【0027】

本発明の算術符号化装置、算術符号化方法、算術符号化プログラム及びプログラムを格納したコンピュータで読み取り可能な記録媒体によれば、H.264/AVCにおけるCABACの処理のサイクル数を削減し、処理の低負荷化、高速化を図ることができる。

30

【発明を実施するための最良の形態】

【0028】

以下、本発明の実施の形態を図面に基づいて説明する。ただし、以下に示す実施の形態は、本発明の技術思想を具体化するための算術符号化装置、算術符号化方法、算術符号化プログラム及びプログラムを格納したコンピュータで読み取り可能な記録媒体を例示するものであって、本発明は算術符号化装置、算術符号化方法、算術符号化プログラム及びプログラムを格納したコンピュータで読み取り可能な記録媒体を以下のものに特定しない。また、本明細書は請求の範囲に示される部材を、実施の形態の部材に特定するものではない。特に実施の形態に記載されている構成部品の寸法、材質、形状、その相対的配置等は特に特定の記載がない限りは、本発明の範囲をそれのみに限定する趣旨ではなく、単なる説明例にすぎない。なお、各図面が示す部材の大きさや位置関係等は、説明を明確にするため誇張していることがある。さらに以下の説明において、同一の名称、符号については同一もしくは同質の部材を示しており、詳細説明を適宜省略する。さらに、本発明を構成する各要素は、複数の要素を同一の部材で構成して一の部材で複数の要素を兼用する態様としてもよいし、逆に一の部材の機能を複数の部材で分担して実現することもできる。

40

【0029】

本明細書において算術符号化装置に接続される操作、制御、入出力、表示、その他の処理等のためのコンピュータ、プリンタ、外部記憶装置その他の周辺機器との接続は、例え

50

ばIEEE1394、RS-232x、RS-422、RS-423、RS-485、USB等のシリアル接続、パラレル接続、あるいは10BASE-T、100BASE-TX、1000BASE-T等のネットワークを介して電氣的に接続して通信を行う。接続は有線を使った物理的な接続に限られず、IEEE802.1x、OFDM方式等の無線LANやBluetooth等の電波、赤外線、光通信等を利用した無線接続等でもよい。さらに検索対象のテキストや画像データの保存やデータベース構築、検索等に関する設定の保存等を行うための記録媒体には、メモリカードや磁気ディスク、光ディスク、光磁気ディスク、半導体メモリ等が利用できる。

(動画像符号化/復号装置)

【0030】

図1に、H.264/AVCの符号化処理により、画像データからビットストリームを得る動画像符号化/復号装置を構成するブロック図の一例を、図1に示す。この図に示す動画像符号化/復号装置100は、メモリ管理を行うDMA1(Direct Memory Access)と、動き探索部2(ME)と、デブロッキングフィルタ3(DF:Deblocking Filter)と、動き補償部4(MC)と、変換部5と、量子化部6と、逆量子化部7と、逆変換部8と、全体制御部9と、可変長符号化部(VLC:Variable Length Coder)である算術符号化/復号装置10と、可変長復号化部11(VLD:Variable Length Decoder)とを備える。算術符号化/復号装置10は、算術符号化技術としてCABCやCAVLCを行う。動画像符号化/復号装置100は、ハードウェアで構成する他、プロセッサでソフトウェア的に処理することも可能である。この場合は、コンピュータにインストールされた算術符号化プログラムにて、算術符号化を行う。なおコンピュータには、汎用のパーソナルコンピュータやワークステーションの他、携帯電話やスマートフォン、PDA、通信機能付カーナビゲーションやデータ通信可能な固定電話等の、動画像を表示可能な機器が適宜利用できる。

【0031】

動画像符号化/復号装置100では、まず符号化処理を行う動画像が入力され、動き探索部2において動き検出処理を行う。この処理で求められた動きベクトルを用い、動き補償部4、変換部5、量子化部6において、動き補償処理、変換処理、量子化処理を行う。量子化処理を行ったブロック係数、および符号化に使用するすべての情報が算術符号化/復号装置10に集められ、ビットストリームを作成する。次のフレームの符号に用いる参照フレームを作成するため、逆符号化部7、逆変換部8、動き補償部4を用いて、その逆処理を行い、参照フレームを作成し、フレームメモリFMに保管する。フレームメモリFMに保管する前に、デブロッキングフィルタ3を用いれば、ブロックノイズを軽減する効果が得られる。

【0032】

この動画像符号化/復号装置100では、符号化以外に復号処理も可能となる。その処理手順は、まず入力されるビットストリームを復号処理を行う可変長復号化部11より解析し、ビットストリームにある画像情報を復号し、参照フレーム生成手順と同じく逆量子化部7、逆変換部8、動き補償部4を用いて画像フレームを復元し、フレームメモリFMに保管する。このように、算術符号化技術には、符号化されたデータの復号も含まれる。本明細書においては、算術符号化装置、方法、プログラム等は、算術符号化のみならず、復号を行う装置、方法、プログラムも含むものとして説明する。

(算術符号化方式)

【0033】

本実施の形態では、H.264/AVCにおけるCABC処理の高速化を図る。CABCは算術符号化方式である。基本的な算術符号化であるエリアス符号化・復号化の概念を、図2に示す。算術符号化では、一般に下限値0以上、上限値1未満の実数直線を考える。この区間を初期のrangeとする。算術符号化においては、優勢シンボル(Most Possible Symbol: MPS)、劣勢シンボル(Least Possible Symbol: LPS)の発生確率に基づいて実数直線を分割する。分割した領域をそれぞれrMPS、rLPSとする

10

20

30

40

50

。その後、符号化出力対象となるシンボルが属する領域を選択する。ここでは仮に r M P S を選択したとする。その後、選択した r M P S を次の r a n g e とし、この r a n g e に対して同様に r M P S、r L P S に分割し、符号化出力対象となる領域を選択する。この動作を符号化出力対象が無くなるまで繰り返し、最終的に特定された領域である r a n g e の範囲内に収まる値を出力値 v a l u e として出力することで符号化が完了する。通常は r a n g e の下限値を出力値 v a l u e として出力する。このような基本的な算術符号化は、区間の r a n g e や入力値である v a l u e が実数値であり、演算精度の問題から現実的ではない。そこで実数直線を有限のビット幅を持つ整数区間とし、特定した区間の幅である r a n g e と特定した区間の下限値である v a l u e に対して正規化処理を導入することで問題を解決している。正規化処理は、左シフト演算により行われ、この正規化処理によって、r a n g e と v a l u e の演算精度を確保している。C A B A C では、複数のコンテキスト変数情報を利用して符号化効率の向上を図っている。

10

【 0 0 3 4 】

次に算術符号化 (A C : Arithmetic Coding) を行う手順を、図 3 に基づいて説明する。C A B A C において用いられる 2 値算術符号化の場合、与えられた 2 値シンボルの発生確率に基づいて、入力信号に応じて実数 0 . 0 から 1 . 0 の間で区間を逐次分割していく。ここで、入力信号 0 0 0 1 0 とすると、" 0 " の発生確率が 0 . 8、" 1 " の発生確率が 0 . 2 となる。このため、入力信号の内 " 0 " が符号化されると、前の区間の下 0 . 8 の部分が、新しい区間として更新される。一方、" 1 " が符号化されると、前の区間の上 0 . 2 が新しい区間として更新される。

20

【 0 0 3 5 】

図 3 の最初のステップ 1 では " 0 " を符号化する。この結果、初期区間の 0 . 0 - 1 . 0 の下側 0 . 8 の部分 (0 . 0 ~ 0 . 8) が新しい区間として更新される。

【 0 0 3 6 】

次にステップ 2 で " 0 " を符号化する。この結果、現在の区間 (0 . 0 ~ 0 . 8) の下側で $0 . 8 \times 0 . 8 = 0 . 6 4$ の部分 (0 . 0 ~ 0 . 6 4) が更新された区間となる。

【 0 0 3 7 】

同様にステップ 3 で " 0 " を符号化する。この結果、現在の区間 (0 . 0 ~ 0 . 6 4) のさらに下側で $0 . 6 4 \times 0 . 8 = 0 . 5 1 2$ の部分 (0 . 0 ~ 0 . 5 1 2) が更新区間となる。

30

【 0 0 3 8 】

次にステップ 4 では " 1 " を符号化する。この結果、現在の区間 (0 . 0 ~ 0 . 5 1 2) の上側で、 $0 . 5 1 2 \times 0 . 2 = 0 . 1 0 2 4$ の部分、すなわち ((0 . 5 1 2 - 0 . 1 0 2 4 = 0 . 4 0 9 6) ~ 0 . 5 1 2) = (0 . 4 0 9 6 ~ 0 . 5 1 2) が更新区間となる。

【 0 0 3 9 】

そしてステップ 5 で再び " 0 " を符号化する。この結果、現在の区間 (0 . 4 0 9 6 ~ 0 . 5 1 2) の下側で $0 . 1 0 2 4 \times 0 . 8 = 0 . 0 8 1 9 2$ の部分 (0 . 4 0 9 6 ~ (0 . 4 0 9 6 + 0 . 0 8 1 9 2)) = (0 . 4 0 9 6 ~ 0 . 4 9 1 5 2) が更新区間となる。

40

【 0 0 4 0 】

最後に、ステップ 6 で符号を決定する。算術符号の符号語は、最終区間を特定する実数値の 2 進表現となる。この例では、最終の更新区間が (0 . 4 0 9 6 ~ 0 . 4 9 1 5 2) となるので、その間に含まれる実数の中で最も 2 進表現が短くなる 0 . 4 3 7 5 をとることができる。0 . 4 3 7 5 の 2 進表現は 0 . 0 1 1 1 になるので、常に 0 となる最初の桁を除いた、小数点以下の " 0 1 1 1 " が符号語として出力される。

(算術符号化装置のブロック図)

【 0 0 4 1 】

次に、C A B A C 処理を行う算術符号化装置の例を、図 4 に示す。この図に示すように算術符号化装置 1 0 A は、多値信号を 2 値シンボルに変換する 2 値化部 1 2 と、符号化す

50

べき2値シンボルの発生確率を周囲の状況すなわちコンテキストに応じて計算・更新するためのコンテキスト計算部14と、これら2値シンボルと2値シンボル発生確率に基づいて算術符号化を行う2値算術符号化部16で構成されている。

【0042】

2値算術符号化部16の詳細を図5に示す。この図に示すように、2値算術符号化部16はrLPSテーブル20と、コンテキスト変更テーブル30と、演算部40と、再正規化部50とを有する。

(再正規化)

【0043】

CABAC内部の2値算術符号化部も、算術符号化にあたっては上記と同様の動作を行う。ただ、実際に区間区間を保持する記憶素子であるレジスタの桁数が有限であるため、出力すべき区間の上位ビットが確定した段階で適時確定ビットを出力し、更新区間の幅を広げる再正規化(Renormalization)の処理が加わる。

【0044】

再正規化の手順を、図6に基づいて説明する。ステップ1で0を符号化した結果、更新区間が(0~0.8)となる。次にステップ2で1を符号化した結果、区間(0.0~0.8)の上側0.2の部分(0.64~0.8)が更新区間となる。この段階で、出力すべき区間の代表値が0.5以上になることが確定する。すなわち、先頭のビットが"1"であることが確定するため、この時点でステップ3として、"1"を出力すると共に、(0.5~1.0)の区間を(0.0~1.0)に拡大するようマッピング(対応付け)する。ここでは、(0.64~0.8)の区間を((0.64-0.5)×2=0.28)~((0.8-0.5)×2=0.6)=(0.28~0.6)に拡大する。このように、処理の途中で(この例ではステップ2で1を符号化した時点で)区間の座標値が0.5(2進表現で0.1)以上となることが判明するので、その時点で小数点第1位の"1"を出力する。同時に、区間から0.5を減算して2倍することにより、区間を実質的に拡張でき、使用可能な桁数を大きくできる。一方、区間の座標値が0.25よりも小さくなると、先頭ビットが"0"であることが判明する。この処理によって、区間の幅を表す実数の桁数が増え続けることを防ぐことができ、有限桁数の演算で符号化を行うことが可能となる。

【0045】

以上の例では、説明のため2値シンボルの発生確率は固定のものとしたが、実際には一動作ごとに適応的に変更させることも可能である。2値算術符号化部に2値シンボルの発生確率を与える役割は、コンテキスト計算部14が行っている。

(CABACの動作)

【0046】

CABACの動作としては、更新区間を更新する、すなわちrange(更新区間)とlow(更新区間以外)に区分する動作として、MPS(Most Possible Symbol)処理とLPS(Least Possible Symbol)処理の2種類がある。上記図3の例では、0を符号化する処理がMPSに該当し、1を符号化する処理がLPSに相当する。

【0047】

MPS処理時には、以下の式によってrangeとlowが決定される。

$$\text{range} = \text{range} - \text{rLPS}$$

$$\text{low} = \text{low}$$

【0048】

一方、LPS処理時には以下の式でrangeとlowが決定される。

$$\text{range} = \text{rLPS}$$

$$\text{low} = \text{low} + \text{range} - \text{rLPS}$$

これらの処理のいずれが行われるかは、コンテキストとrangeによって決定される。コンテキストは、2値シンボルの発生確率を示しており、シンボルの処理時に更新される。

【 0 0 4 9 】

コンテキストは図 4 に示すコンテキスト計算部 1 4 で保持、更新される。コンテキスト計算部 1 4 は、2 値シンボルの発生確率を複数保持しており、現在の符号化対象や周囲の状況すなわちコンテキストに応じて発生確率を切り替え、2 値算術符号化部 1 6 に与える。2 値シンボルの発生確率は、発生確率テーブルに保持され、発生確率が高い 0、1 いずれかのシンボル (M P S : Most Probable Symbol : 発生確率が最も高いシンボル) と、その発生確率テーブル番号 (pStateIdx) とが対応付けて保持される。発生確率テーブル番号を指定すると、対応する発生確率が得られる。

(従来 の C A B A C 処理)

【 0 0 5 0 】

次に、C A B A C 処理を行う従来の算術符号化装置において、2 値算術符号化部 1 6 が、コンテキスト計算部 1 4 から与えられた発生確率 (コンテキスト) に従って、2 値化部 1 2 から与えられる 2 値シンボルの符号化を行い、区間を更新する手順を、図 7 のブロック図及び図 8 の動作図に基づいて説明する。

【 0 0 5 1 】

図 7 に示す算術符号化装置 2 0 0 は、演算部 4 0 と、r L P S テーブル 2 0 と、コンテキスト更新部 7 0 とを備える。また算術符号化装置 2 0 0 は、入力端子として、区間を入力する r a n g e 入力及び / 又は l o w 入力、2 値列入力、コンテキスト入力を備えている。r a n g e 入力、l o w 入力はそれぞれ、演算部 4 0、r L P S テーブル 2 0 に接続される。また 2 値列入力は演算部 4 0、r L P S テーブル 2 0、コンテキスト更新部 7 0 に接続される。さらにコンテキスト入力は、r L P S テーブル 2 0、コンテキスト更新部 7 0 に接続される。ここで r L P S テーブル 2 0 は、指定された r a n g e とコンテキストに応じて、r L P S を選択するテーブルである。また演算部 4 0 は再正規化部を備え、再正規化の際に符号ビットを生成し、演算部 4 0 から出力符号を出力する。コンテキストは、r a n g e、l o w の更新後に更新される。コンテキスト更新部 7 0 は、前段の処理結果に基づいてコンテキストを更新して出力する。

【 0 0 5 2 】

演算部 4 0 は、算術符号化演算を行う。ここでは、入力された区間 r a n g e 及び / 又は l o w、これに対応する r L P S を r L P S テーブル 2 0 を参照して決定し、これらに基づいて区間を更新し、r a n g e ' と l o w ' を出力する。

【 0 0 5 3 】

この処理を、図 8 の動作図に基づいて説明する。ここでは、所定の区間に対して、(サイクル 1) M P S 処理と、(サイクル 2) L P S 処理及び再正規化を行う処理を説明している。すなわち、まずサイクル 1 で、r a n g e と l o w が決定されている区間に対して、2 値シンボル A 及び対応するコンテキスト X の場合に演算部 4 0 が M P S 処理して符号化を行い、区間を r a n g e ' と l o w ' に更新する。ここでは、

$$r a n g e ' = r a n g e - r L P S、$$

$$l o w ' = l o w$$

となる。一方でコンテキスト更新部 7 0 がコンテキスト X を更新する。そしてサイクル 2 で、この r a n g e ' と l o w ' を 2 値シンボル B 及び対応するコンテキスト Y の場合に演算部 4 0 が L S P 処理で符号化を行って、区間を r a n g e " と l o w " に更新する。ここでは

$$r a n g e " = r L P S '、$$

$$l o w " = l o w ' + r a n g e ' - r L P S '、$$

となる。また一方、コンテキスト更新部 7 0 がコンテキスト Y を更新する。

【 0 0 5 4 】

さらに再正規化を行って区間を r a n g e " ' と l o w " ' に拡大する。この結果、

$$r a n g e " ' = (r a n g e " - 0 . 5) \times 2$$

$$l o w " ' = (l o w " - 0 . 5) \times 2$$

となる。さらに演算部 4 0 が出力符号 " 1 " を出力する。このように従来の C A B A C 処

10

20

30

40

50

理では、上記の処理を完了するまでに2サイクルを要した。

【0055】

一方で近年の画像の高精細化に伴ってH.264/AVCの符号化処理をリアルタイムで行うなど、高速化の要求が高く、CABAC処理の演算量を削減することが求められている。しなしながら、上記の構成ではrange、lowの値が前の処理に依存するため、演算をパイプライン化できないという問題があった。すなわち、隣接するシンボルの算術符号化においては、2シンボル目の処理にはrange、lowだけでなくrLPSの値までも前の処理(1シンボル目の処理)に依存する。このため、1サイクル間に2シンボル以上の処理は困難とされていた。

【0056】

言い換えると、従来、CABACの実行速度の高速化手法としては、CABAC装置の各処理部の処理やメモリアクセス、符号化器部分の動作並列化等に着目し、如何に効率よくデータ処理を行うかという手法を主眼として、1シンボル/1サイクルの処理の実現が研究されている。しかしながら、CABACのシンボル処理は、常に過去のシンボルの出現確率に依存する特徴があり、各シンボルの処理時に更新が必要なコンテキスト、rLPSは、直前のシンボルの処理結果を得なければならない。このため従来は、前段の処理に依存する性質上、演算処理のハードウェアを増やすことでは処理の効率化は望めないと考えられており、十分に満足できる高速化が実現されていないにもかかわらず、1シンボル/1サイクルの処理速度が高速化の限界であると理解されていた。

【0057】

これに対し本発明では、シンボルの出現確率の演算手順を分析し、シンボル間の依存性の詳細関係を解明した。これにより、状況によりシンボル間の依存性を失くし、事前に次サイクルの処理を実行し、テーブルの更新も同時に実現できる手法を開発するに至った。(2サイクル予測)

【0058】

本実施の形態では、2シンボルの処理に必要な情報を一括して入力し、1サイクル間に2シンボルの処理を行うため、コンテキスト更新テーブル、rLPSテーブル、演算部をそれぞれ2つ設けている。ここではrangeとlowを演算するために必要なコンテキスト、すなわち2値列とシンタックス要素と周囲の状況(コンテキスト)に基づいて、生成された2値列に使用するコンテキストを決定するためのrLPSテーブルを予め作成し、処理時にはこれを参照することで先の演算の予測を可能としている。さらにrLPSテーブルを2つ用意することで、2サイクル分の処理を行うための異なるrLPSを、同時に参照可能とした。例えば、一方のテーブルをMPS処理用のLPSを記録した第1rLPSテーブルとし、他方をLPS処理用のLPSを記録した第2rLPSを保持したテーブルとしており、1サイクル中にMPS処理、LPS処理両方に対応できる。また各rLPSテーブルはマルチポート(デュアルポート)メモリを利用し、2つのアクセスポートから同時に書き込みと読み出しを可能としている。これにより、一方のアクセスポートからデータを読み書きしているときに、他方のアクセスポートを使用してデータの読み出しなどが行なえる。これにより、2つのプロセッサが同じメモリに同時にアクセス可能となり、2サイクル分の処理を1サイクル内で完了でき、処理サイクル数を低減してCABAC処理の高速化を図っている。

【0059】

なおコンテキストをメモリから2つ同時に読み出し、且つ2つ同時に書き込むには、更新コンテキストの内容により、最大で従来の4倍程度のメモリ量が必要になると見込まれる。またメモリ量を従来の2倍のみ使用した場合には、同じコンテキストの更新が必要となる際に2サイクルの処理を要し、処理速度の低減になる。同様に3シンボルの処理も可能であるが、コンテキストのメモリ量は従来の6倍程度となる。このように、パフォーマンスとハードウェア仕様のバランス、及び要求される処理能力や速度などに応じて、適切な構成に設計する必要がある。好ましくは、一括に入力する情報は2シンボル分とし、2サイクル予測と、後述する連続MPS処理予測を併用して、これらを切り替えることによ

10

20

30

40

50

り、画像にあまり依存せず、高い処理速度を保つことができる。以下、順次説明する。

(実施の形態1)

【0060】

図9に、本実施の形態1に係る算術符号化装置を示す。この図に示す算術符号化装置は、多値信号を2値シンボルに変換する2値化部112と、符号化すべき2値シンボルの発生確率を周囲の状況(コンテキスト)に応じて計算・更新するためのコンテキスト計算部114と、これら2値シンボルと2値シンボル発生確率に基づいて算術符号化を行う算術符号化部116と、コンテキストを保持する第1メモリ部117と、range及びlowを保持する第2メモリ部118と、2サイクル予測を行う予測部119と、出力バッファ121とを備える。この算術符号化装置は、多値信号及び/又は符号化対象のシンタクス要素を2値化部112で入力すると、2値列に変換して算術符号化部116と予測部119に送られる。また必要に応じて周囲の状況(コンテキスト)もコンテキスト計算部114に入力される。そして予測部119でrangeの予測を行う一方、算術符号化部116で算術符号化を行い、得られた結果を符号ビットとして各々出力バッファ121に出力する。なお、必要に応じて入力側にバッファを設けることも可能であることは言うまでもない。

10

(第1メモリ部117)

【0061】

第1メモリ部117は、コンテキストを保持するコンテキストテーブルとして機能する。コンテキストテーブルは必要に応じて随時更新される。このため第1メモリ部117には更新が可能なメモリ、例えばRAMなどが使用される。第1メモリ部117の例を図10に示す。この図に示すように第1メモリ部117内にはコンテキスト#1、#2、#3、・・・が保持され、各コンテキストが2値列(0,1)と対応付けられる。この際、1のコンテキストに複数の2値列を割り当てることも可能である。

20

【0062】

第2メモリ部118は、コンテキストを保持する第2コンテキスト更新テーブルとして機能する。range、low値の保持には、レジスタなどが使用できる。

【0063】

また、図11の変形例に示すように、第1メモリ部と第2メモリ部を共通のメモリ部117Aとして構成し、コンテキストとrange、lowを一のメモリ内に各々保持することも可能である。

30

【0064】

第1メモリ部117にはデュアルポート型メモリを使用する。これによって、コンテキストテーブルに同時にリード/ライトを行うことが可能となる。

(算術符号化部116)

【0065】

図9の算術符号化部116の詳細を、図12のブロック図及び図13の動作図に基づいて説明する。図12に示す算術符号化部116は、演算部140と、第1rLPSテーブル120と、第2rLPSテーブル122と、符号ビット生成部160と、コンテキスト更新部170とを備える。また算術符号化部116は、入力端子として、区間を入力するrange入力及び/又はlow入力、2値列入力、コンテキスト入力としてコンテキストX入力、コンテキストY入力を備えている。range入力は、演算部140、第1rLPSテーブル120、第2rLPSテーブル122、符号ビット生成部160に接続される。一方low入力は、演算部140、第2rLPSテーブル122、符号ビット生成部160に接続される。また2値列入力は演算部140、第2rLPSテーブル122、符号ビット生成部160、コンテキスト更新部170に接続される。さらにコンテキストX入力は、第1rLPSテーブル120、第2rLPSテーブル122、符号ビット生成部160、コンテキスト更新部170に接続される。一方コンテキストY入力は、第2rLPSテーブル122、符号ビット生成部160、コンテキスト更新部170に接続される。

40

50

【 0 0 6 6 】

第1 r L P Sテーブル1 2 0、第2 r L P Sテーブル1 2 2は、それぞれ入力された r a n g eとコンテキストに応じて、r L P Sを決定するテーブルである。テーブルを2つ備えたことで、1サイクル中に2ビットの読み出しや書き込みを行うことが可能となり、高速な処理が見込まれる。また一方、いずれのテーブルに最新のデータが保持されているかを管理し、必要に応じてテーブルを切り替える制御が必要となる。このため、図9の算術符号化部1 1 6はテーブル管理手段を備えている。

【 0 0 6 7 】

符号ビット生成部1 6 0は、再正規化の際に出力符号を出力する。さらにコンテキスト更新部1 7 0は、前段の処理結果に基づいてコンテキストを更新して出力する。演算部1 4 0は、入力された区間 r a n g e及び/又は l o w、これに対応する r L P Sを r L P Sテーブル2 0を参照して決定し、これらに基づいて区間を更新し、 r a n g e 'と l o w 'を出力する。

10

【 0 0 6 8 】

この処理を、図13の動作図に基づいて説明する。図13は図8と同様、所定の区間に対してM P S処理、L P S処理、再正規化を行う処理を説明している。ここでは r a n g eと l o wが決定されている区間に対して、2値シンボルA及び対応するコンテキストXの場合に演算部1 4 0がM P S処理して符号化を行い、区間を r a n g e 'と l o w 'に更新する処理と、この r a n g e 'と l o w 'を2値シンボルB及び対応するコンテキストYの場合に演算部1 4 0がL P S処理で符号化を行って、区間を r a n g e "と l o w "に更新する処理、及び再正規化を行って区間を r a n g e " 'と l o w " 'に拡大する処理と、コンテキスト更新部1 7 0がコンテキストX、コンテキストYをそれぞれ更新する処理を、1サイクルで行う。すなわち、演算部1 4 0は第1 r L P Sテーブル1 2 0を参照してM P S時の r L P Sを、第2 r L P Sテーブル1 2 2を参照してL P S時の r L P S 'を、各々取得し、以下の計算を行う。

20

$$r a n g e " ' = (r L P S ' - 0 . 5) \times 2$$

$$l o w " ' = (l o w + r a n g e - r L P S - r L P S ' - 0 . 5) \times 2$$

【 0 0 6 9 】

また同じサイクル内で符号ビット生成部1 6 0が出力符号 " 1 "を出力し、コンテキスト更新部1 7 0はコンテキストX、コンテキストYをそれぞれ更新する。これによって、1サイクルでM P S処理、L P S処理、再正規化を完了させることができる。

30

【 0 0 7 0 】

次に、図14に、他の構成に係る算術符号化部3 0 0の例を示す。この図に示す算術符号化部3 0 0は、1サイクル中で2シンボルを処理するために、1シンボル目処理用の第1算術符号化手段3 0 1と、2シンボル目処理用の第2算術符号化手段3 0 2と、符号ビット生成部3 6 0を備える。第1算術符号化手段3 0 1は、第1 r L P Sテーブル3 2 0と、第1コンテキスト更新テーブル3 3 0と、第1演算部3 4 0と、第1再正規化部3 5 0とを備える。また第2算術符号化手段3 0 2は同様に、第2 r L P Sテーブル3 2 2と、第2コンテキスト更新テーブル3 3 2と、第2演算部3 4 2と、第2再正規化部3 5 2とを備える。この算術符号化部3 0 0は、2値列としてシンボルA及びコンテキストXを入力すると、コンテキストX及び r a n g eに対応した r L P Sを第1 r L P Sテーブル3 2 0を参照して求め、この値とシンボルAとで第1演算部3 4 0は算術符号化を行い、 r a n g eと l o wを更新する。さらに必要に応じて第1再正規化部3 5 0で再正規化を行い、 r a n g e 'と l o w 'を第2算術符号化手段3 0 2に送出すると共に、符号ビットを符号ビット生成部3 6 0に出力する。また第1コンテキスト更新テーブル3 3 0を更新し、コンテキストX 'を出力する。

40

【 0 0 7 1 】

一方、第2算術符号化手段3 0 2は、コンテキストY及び r a n g e 'に対応した r L P Sを第2 r L P Sテーブル3 2 2を参照して求め、この値と r a n g e 'と l o w 'に基づいて第2演算部3 4 2で算術符号化を行い、 r a n g e 'と l o w 'を更新する。さ

50

らに必要に応じて第2再正規化部352で再正規化を行い、range”とlow”を出力すると共に、符号ビット生成部360に符号ビットを出力する。符号ビット生成部360は、出力符号ビットを出力する。またシンボルB及びコンテキストYに基づいて第2コンテキスト更新テーブル332を更新し、コンテキストY'を出力する。

【0072】

以上の処理は、1サイクル中で終わることができる。すなわち、rangeとlow、コンテキストXに基づいて第1算術符号化手段301で1シンボル目のシンボルAを算術符号化すると共に、得られたrange'とlow'、及びコンテキストYに基づいて第2算術符号化手段302が2シンボル目のシンボルBを算術符号化する。このように本実施の形態では、2値列とシンタックス要素と周囲の状況(コンテキスト)に基づいて、生成された2値列に使用するコンテキストを決定するrLPSテーブルを予め作成しておくと共に、rLPSテーブルを2つ用意することで、1サイクル中に演算部は各々のテーブルから必要なデータを読み出し、更新することができ、2サイクル分の演算を1サイクルで完了することができる。この結果、高い符号化効率を達成する一方で処理時間の長さの問題となっていたCABACの処理サイクル数を削減することが可能となり、もって処理の高速化を図り、高解像度アプリケーションのリアルタイム符号化等に実装可能なシステムを構築できる。この手法は、汎用のCPUやMPUでソフトウェア的に実装することも可能であるが、好ましくは専用のハードウェアを構築して実行させることで、より効率のよい処理が可能となる。

(2サイクル予測を行う算術符号化部の変形例)

【0073】

図9のブロック図に示す算術符号化装置の構成では、算術符号化部116で算術符号化演算を行う。算術符号化部116の具体的な構成としては、例えば図5のブロック図に示した構成が利用できる。ここで、コンテキスト更新テーブルを各々ROMで構成すると、各々がシンボルに応じてコンテキストを決定するため、ROMのサイズを大きくする必要がある。また、ROMではデータのアクセスに1サイクルを要するため、遅延が生じる。そこで、一方のコンテキスト更新テーブルを組み合わせ回路で構成することでROMのサイズを小さくでき、安価に構成できる。さらに組み合わせ回路では、演算論理の結果はそのサイクル中に生成されるので、余分なサイクルを要さず、処理を高速化できる利点を得られる。組み合わせ回路を用いた算術符号化部216の例を、図29に示す。

【0074】

この図に示す算術符号化部216は、コンテキストメモリ部217と、図29においてコンテキストメモリ部217の第1の出力側に接続された第1コンテキスト更新テーブル230と、第2の出力側に接続された第2コンテキスト更新テーブル232と、コンテキストメモリ部217の第1の出力側に接続された第1rLPSテーブル220と、第1コンテキスト更新テーブル230とコンテキストメモリ部217の加算器出力と接続された第2rLPSテーブル222と、rangeの値を保持するためのrangeメモリ部218と、rangeメモリ部218及び第1rLPSテーブル220の出力側に接続された第1選択器224と、第1選択器224及びrangeメモリ部218に接続された第1range更新部240と、第2rLPSテーブル222及び第1range更新部240の出力側に接続された第2選択器226と、第2選択器226及び第1range更新部240の出力側に接続され、且つ出力側をrangeメモリ部218に接続した第2range更新部242と、lowの値を保持するlowメモリ部219と、lowメモリ部219、第1range更新部240及び第1選択器224の出力側に接続された第1low更新部250と、第1low更新部250、第2range更新部242及び第2選択器226の出力側に接続され、且つ出力側をlowメモリ部219に接続した第1low更新部252とを備える。

【0075】

なお図29は、処理工程とデータの流れをブロック図的に表現した図であるため、実際のハードウェア的な結線とは必ずしも一致しない。例えばrange値については、ra

10

20

30

40

50

rangeメモリ部218 第1rangeテーブル 第2rangeテーブル rangeメモリ部218にループして、rangeが順次更新される様子を示しているが、実際には第1rangeテーブル及び第2rangeテーブルがそれぞれrangeメモリ部218に接続され、各々のrangeテーブルで更新したrange値をrangeメモリ部218に書き込むことができる。またコンテキストメモリ部217が重複して表示されているが、両者は同じものであり、図29において左側のコンテキストメモリ部217はリード時、中央のコンテキストメモリ部217はライト時の動作をそれぞれ示している。

【0076】

コンテキストメモリ部217は、図9においてコンテキストを保持する第1メモリ部117に相当する。ここでは、シンボルAとこれに対応するコンテキストX、シンボルBとこれに対するコンテキストYの4つの入力、及び更新されたコンテキストX'、Y'2つの出力を備えるデュアルポートRAM等で構成される。このコンテキストメモリ部217は、更新されたコンテキストを保持すると共に、指定されたアドレスに従い該当するコンテキストの読み込みを行い、さらに第1、第2コンテキスト更新テーブル232によって更新された2つのコンテキストX'、Y'の書き込みを各々行う。ここで第2コンテキスト更新テーブル232は、ROMで構成されるが、一方第1コンテキスト更新テーブル230は組み合わせ回路で構成される。一方のコンテキスト更新テーブルを組み合わせ回路で構成することで、第2段階でコンテキストX、Yを同時に更新できるので、1サイクルで2シンボルの処理を行う高速化に大きく資する。

【0077】

なお、図29に示す算術符号化部216の例において、第1コンテキスト更新テーブル230及び第2コンテキスト更新テーブル232の両方を組み合わせ回路で構成することもできる。この場合は、一層の高速化を図ることができる。なお、コンテキスト更新テーブルに大量のデータを記憶する場合は、組み合わせ回路で実現すると大面積化と高コスト化となるため不適となる。ただ、保存すべきデータ量が少ない場合は、組み合わせ回路で実装しても面積がROMの場合に比べさほど大きくならないため、両方を組み合わせ回路で構成することが好ましい。

【0078】

第1及び第2rLPSテーブル222は、各々ROMで構成され、入力されたコンテキストに従い、rLPSを出力する。ROMに保持されたrLPSテーブルのデータは、コンテキストに応じて4つに区分されているため、各rLPSテーブルが4つの出力を備える。各々のrLPSテーブルから出力されたrLPSは、第1及び第2選択器226にそれぞれ入力される。第1、第2選択器226は、各々入力されるrangeに従い、該当するrLPSを選択して、第1、第2range更新部242に出力する。第1、第2range更新部242は、rangeの値を更新してrangeメモリ部218のrange値を書き換える。同様にlowメモリ部219は、low値を保持する。これらrangeメモリ部218やlowメモリ部219は、レジスタで構成できる。また第1、第2low更新部252は、再正規化部の機能を実現しており、range及び再正規化の回数に従い再正規化してlowを更新する。このようにして第1low更新部250はシンボルAの算術符号化された符号ビット1を出力し、第2low更新部252はシンボルBの算術符号化された符号ビット2を出力する。

【0079】

なお図29の例では、各処理を行う部材を機能的に表現したものであって、実際のハードウェア構成においては、一の部材で複数の機能を実現することができる。例えば、コンテキストメモリ部217は図9において第1メモリ部117に、rangeメモリ部218及びlowメモリ部219は、第2メモリ部118に相当する。また、第1range更新部240及び第1low更新部250は、図14に示す第1演算部に、第2range更新部242及び第2low更新部252は、第2演算部に、各々相当し、さらにこれらを適宜纏めることもできる。

(DSEの算術符号化処理工程)

10

20

30

40

50

【 0 0 8 0 】

以上の算術符号化部 2 1 6 で 1 サイクル中に 2 シンボルの算術符号化演算を行う処理は、図 2 9 及び図 3 0 のフローチャートに示すように、4 段階に分けることができる。すなわち、ステップ S 1 で示す第 1 段階では、各コンテキストの読み込みが行われる。具体的には、コンテキストメモリ部 2 1 7 で、アドレスに従いシンボル A、B と対応するコンテキスト X、Y の読み出しを行う。図 2 9 において、左側のコンテキストメモリ部 2 1 7 はリード状態を示している。

【 0 0 8 1 】

次にステップ S 2 で示す第 2 段階で、各コンテキストの更新と各 r L P S の読み込みが行われる。具体的には、コンテキストメモリ部 2 1 7 の第 1 の出力から第 1 コンテキスト更新テーブル 2 3 0 がコンテキスト X を読み込み、コンテキスト X ' に更新する。また、更新されたコンテキスト X ' を第 2 コンテキスト更新テーブル 2 3 2 が参照し、他方でコンテキストメモリ部 2 1 7 の第 2 の出力から得られたコンテキスト Y をコンテキスト Y ' に更新する。ここで、コンテキスト更新テーブルを 2 つの ROM とせず、一方を組み合わせた回路で構成した第 1 コンテキスト更新テーブル 2 3 0 とし、他方を ROM で構成された第 2 コンテキスト更新テーブル 2 3 2 とすることで、処理の高速化と小面積化を両立させ、2 つのシンボル A、B のコンテキスト X、Y を同時に更新することが可能となる。また上述の通り、すべてのコンテキスト更新テーブルを組み合わせた回路で構成することで、一層の高速化が図られる。

【 0 0 8 2 】

また一方で、第 2 段階において更新前のコンテキスト X に基づき、シンボル A の r L P S を r L P S テーブルを参照して求める。すなわち、コンテキストメモリ部 2 1 7 の第 1 の出力から得られるコンテキスト X に基づき、第 1 r L P S テーブル 2 2 0 が r L P S を第 1 選択器 2 2 4 に出力する。他方、第 1 コンテキスト更新テーブル 2 3 0 で得られるコンテキスト X ' 及びコンテキストメモリ部 2 1 7 の第 2 の出力から得られるコンテキスト Y の AND 出力が第 2 r L P S テーブル 2 2 2 に送られ、同様に更新前のコンテキスト Y の r L P S が求められ、第 2 選択器 2 2 6 に出力される。

【 0 0 8 3 】

さらにステップ S 3 で示す第 3 段階で、更新されたコンテキストの書き込みと r a n g e の更新が行われる。まず、第 1 コンテキスト更新テーブル 2 3 0 及び第 2 コンテキスト更新テーブル 2 3 2 から各々出力されるコンテキスト X '、Y ' がコンテキストメモリ部 2 1 7 に書き込まれる。図 2 9 において、中央下段のコンテキストメモリ部 2 1 7 はライト状態を示している。

【 0 0 8 4 】

一方で、第 1 選択器 2 2 4 は、r a n g e メモリ部 2 1 8 を参照して得られる r a n g e に従い、シンボル A の該当する r L P S を選択して、第 1 r a n g e 更新部 2 4 0 に出力する。第 1 r a n g e 更新部 2 4 0 は、r a n g e メモリ部 2 1 8 に保持される現在の r a n g e を、選択された r L P S に応じて r a n g e ' に更新すると共に、次段のシンボル B に相当する r a n g e を決定するために第 2 r a n g e 更新部 2 4 2 に出力する。また一方で、更新された r a n g e ' に応じて、再正規化の回数を決定し、第 1 l o w 更新部 2 5 0 に出力する。

【 0 0 8 5 】

第 2 r a n g e 更新部 2 4 2 は、第 1 r a n g e 更新部 2 4 0 で更新された r a n g e ' を、第 2 選択器 2 2 6 で選択された r L P S に従い、r a n g e " に更新し、r a n g e メモリ部 2 1 8 の r a n g e 値を r a n g e " に更新する。一方、更新された r a n g e " に応じて再正規化の回数を決定し、第 2 l o w 更新部 2 5 2 に出力する。

【 0 0 8 6 】

最後にステップ S 4 で示す第 4 段階で、l o w の更新が行われる。第 1 l o w 更新部 2 5 0 は、r a n g e ' 及び再正規化の回数に応じて、l o w メモリ部 2 1 9 に保持された l o w 値を l o w ' に更新すると共に、シンボル A の符号ビット 1 を出力する。一方第 2

10

20

30

40

50

low更新部252は、range”及び再正規化の回数に応じてlow’をlow”に更新し、lowメモリ部219のlow値をlow”に更新する。また、シンボルBの符号ビット2を出力する。

【0087】

以上のようにして、2つのコンテキスト更新テーブルの一方をROM、他方を組み合わせ回路で構成して、1サイクルで2シンボルの算術符号化を安価に実現できる。

(連続MPS処理予測)

【0088】

一方で、CABAC処理による2値化の特徴として、同じシンボルを連続して生成しやすい点が挙げられる。またCABAC処理に含まれるMPS処理とLPS処理の内、MPS処理の特徴として、lowは変化せずrangeの値のみ緩やかに減少するという傾向がある。そこで、2値列がMPSの連続である場合、この連続数とrange及びコンテキストから、数サイクル先までの結果を予測することが可能となる。

【0089】

この処理の手順を、図15の動作図に基づいて説明する。ここではコンテキストXが一定値で、MPS処理が連続して4回続くものとする。まず第1サイクルでrange0がrange1に更新されるとすると、このときのMPS処理で使用するrLPSがrLPS1であれば、

$$\text{range } 1 = \text{range } 0 - \text{rLPS } 1$$

$$\text{low } 0 = \text{low } 1$$

となる。また次の第2サイクルでrange1がrange2に更新されるとすると、このときのMPS処理で使用するrLPSがrLPS2であれば、

$$\text{range } 2 = \text{range } 1 - \text{rLPS } 2$$

$$= \text{range } 0 - \text{rLPS } 1 - \text{rLPS } 2$$

$$\text{low } 2 = \text{low } 1 = \text{low } 0$$

となる。さらに第3サイクルでrange2がrange3に更新され、このときrLPS3であれば、

$$\text{range } 3 = \text{range } 2 - \text{rLPS } 3$$

$$= \text{range } 0 - \text{rLPS } 1 - \text{rLPS } 2 - \text{rLPS } 3$$

$$\text{low } 3 = \text{low } 2 = \text{low } 1 = \text{low } 0$$

となり、最後に第4サイクルでrange3がrange4に更新され、このときrLPS4であれば、

$$\text{range } 4 = \text{range } 3 - \text{rLPS } 4$$

$$= \text{range } 0 - \text{rLPS } 1 - \text{rLPS } 2 - \text{rLPS } 3 - \text{rLPS } 4$$

$$\text{low } 4 = \text{low } 3 = \text{low } 2 = \text{low } 1 = \text{low } 0$$

となる。このように同一コンテキストを用いたMPSの連続した処理は、range、コンテキストX、連続数から演算することができるので、予め演算した結果をテーブルに記録しておくことで、上記のパラメータから数サイクル先まで予測可能となる。これを実現するために、MPS連続数テーブルを用意する。

(MPS連続数テーブル)

【0090】

図16に、コンテキストが一定の場合のMPS連続数テーブルの例を示す。この図に示すように、MPS連続数テーブルは、range、コンテキスト、MPS連続数に応じたrangeを表として保持している。このため、これらのパラメータでテーブルを参照すれば、連続したMPS処理後のrangeの値を取り出すことができる。図15の例では、初期値であるrangeをrangeAmコンテキストX(一定)、MPS連続数4であるから、range予測値は、図16のMPS連続数テーブルにおいてEの位置に格納された値となる。このように、テーブルの参照のみによって数サイクル先の更新区間を求めることができるので、特にMPS処理が連続する場合には極めて大幅な演算量の低減が見込まれ、処理数低減による高速化が図られる。

10

20

30

40

50

(再正規化による変化)

【0091】

以上のMPS処理では、LPS処理が生じない間、すなわちMPS処理が連続する間はlowは一定である。しかしながら、図17(a)に示すように再正規化を行うとlowの値が変化する。ただ、再正規化の値はrangeによって決まる。このため、rangeの値と再正規化を行う回数が判れば、lowの値を求めることができる。このように、再正規化によって変化するlowについてもテーブル化することが可能となる。

【0092】

再正規化を考慮した連続MPS処理予測部400のブロック図を図18に示す。この図に示す連続MPS処理予測部400は、rangeテーブル420と、コンテキスト更新テーブル430と、再正規化部450を備える。この連続MPS処理予測部400は、rangeとMPS連続数、コンテキストに応じてrangeテーブル420を参照し、rangeをrange'に更新する。一方で再正規化回数とlowに基づいて、再正規化部450がlow'と符号ビットを出力する。さらに、MPS連続数とコンテキストに基づいて、コンテキスト更新テーブル430を参照し、コンテキストを更新する(context context')。

10

【0093】

再正規化を考慮したrange予測の例を、図17(b)に示す。この図に示すように、MPS連続数と再正規化の回数に応じて、rangeの値を予測でき、さらにコンテキストを更新できる。

20

(コンテキストの変化)

【0094】

また上記の例ではコンテキストが一定としたが、コンテキストが途中で切り替わる場合もあり得る。上記の手法では同一シンボルに対して同一コンテキストを使用することを前提としていたため、コンテキストが変更されるとそれ以降の変化を予測できないこととなる。そこで、rangeテーブルとコンテキスト更新テーブルとを各々2つ用意することで、このようなコンテキストの変化にも対応可能な連続MPS処理予測を実現する。

【0095】

図19は、連続するMPS処理において、コンテキストが4サイクルまではコンテキストX、続く2サイクルはコンテキストYに切り替わる例を示している。この場合も、コンテキスト毎にテーブル化することが可能である。図20に、range、コンテキスト(可変)、MPS連続数に応じたrange予測値を記録したMPS連続数テーブルの例を示す。図19の例では、初期値であるrangeをrange A、コンテキストXでのMPS連続数が4、その時点でのrangeがrange Eで、その後のコンテキストYでのMPS連続数が2であるため、range予測値は、図20のMPS連続数テーブルにおいてEの位置に格納された値と、Iの位置に格納された値の和E+Iとなる。このように、コンテキストが変化する場合であっても、同様にテーブルの参照によって容易に更新区間を求めることができるので、大幅な処理の低減と高速化が図られる。

30

【0096】

コンテキストが変化する場合に連続MPS処理予測を実現する連続MPS処理予測部400Bを構成するブロック図の例を図21に示す。この図に示す連続MPS処理予測部400Bは、2つの予測テーブル420A、420B(MPS連続数テーブル)を設けることで、rangeとコンテキスト、MPS連続数が入力されると、これらのパラメータに応じたrange予測値が参照され、対応する値としてrange"が出力される。また予測テーブル420A、420B内のコンテキストは適宜更新される。この例ではコンテキストをX、Yの2種類とした例を説明したが、さらに多くのコンテキストに対応させた複数のテーブルを用意することも可能であることは言うまでもない。

40

【0097】

図22に、より詳細な連続MPS処理予測部500のブロック図を示す。この図に示すように、連続MPS処理予測部500は第1コンテキスト更新テーブル530と、第2コ

50

ンテキスト更新テーブル532と、第1rangeテーブル520と、第2rangeテーブル522と、再正規化テーブルを有する一の再正規化部550を備える。この連続MPS処理予測部500は、rangeとMPS連続数X(図19の例では4)、コンテキストA(図19ではコンテキストX)に基づいて、第2rangeテーブル522を参照し、range'(rangeE)を求める。一方、コンテキストB(図19ではコンテキストY)、MPS連続数Y(図19では2)、第2rangeテーブル522で得られたrange'に基づいて、第1rangeテーブル520を参照し、range''を求める。一方で、第1rangeテーブル520及び第2rangeテーブル522の再正規化回数と、lowの値に基づいて、再正規化部550が再正規化テーブルを参照してlow'と、再正規化による符号ビットを出力する。このように、一の再正規化部550でコンテキストの異なる再正規化を加算して、適切な符号ビットを出力することができ、回路の簡素化に寄与する。さらにまた、コンテキストAを、MPS連続数に基づき第2コンテキスト更新テーブル532を参照してコンテキストA'に更新し、コンテキストBをMPS連続数に基づいて第1コンテキスト更新テーブル530を参照し、コンテキストB'に更新する。また上述の通り、コンテキスト変更数を増やすことも可能である。
(連続MPS処理予測を行う算術符号化部の変形例)

10

【0098】

また上記図29で説明したのと同様に、連続MPS処理予測においても、コンテキスト更新テーブルを各々ROMで構成する他、より安価で安定した高速化を得るために、一方を組み合わせ回路で構成したり、両方を組み合わせ回路で構成することもできる。このような算術符号化部316の例を、図31に示す。図31も処理工程をブロック的に表現した図であるため、コンテキストメモリ部217Bが重複して表示されているが、両者は同じものである。

20

【0099】

この図に示す算術符号化部316も、第1及び第2コンテキスト更新テーブル230B、232Bの出力側に接続されたコンテキストメモリ部217Bと、コンテキストメモリ部217Bの第1の出力側に接続された第1コンテキスト更新テーブル230Bと、第2の出力側に接続された第2コンテキスト更新テーブル232Bとを備え、これらの構成は図29とほぼ同様である。すなわち、第1コンテキスト更新テーブル230Bは組み合わせ回路で構成され、第2コンテキスト更新テーブル232BはROMで構成される。あるいは、第1コンテキスト更新テーブル230B及び第2コンテキスト更新テーブル232Bのいずれも、組み合わせ回路で構成してもよい。特に保存すべきデータ量が少ない場合は、実装面積も大きくならないので、より高速化が図られる組み合わせ回路のみでの構成が好ましい。

30

【0100】

また算術符号化部316は、rangeの値を保持するrangeメモリ部218Bと、lowの値を保持するlowメモリ部219Bも備え、これらもレジスタ等で構成できる。これらの部材は、図29とほぼ同様に構成できるので、詳細説明は省略する。

【0101】

一方で算術符号化部316は、シンボルA(MPS)の連続数m及び対応するコンテキストXを入力し、rangeメモリ部218Bから読み出したrangeをrange'に更新する第1rangeテーブル520Bと、シンボルB(MPS)の連続数n及び対応するコンテキストYを入力し、第1rangeテーブル520Bのrange'を更新してrange''を出力する第2rangeテーブル522Bを備える。これらrangeテーブルは、MPS連続数に基づき、rangeを予測する。さらに算術符号化部316は、図29と同様に第1rangeテーブル520Bで更新されたrange'と再正規化の回数に基づいて再正規化を行い、lowメモリ部219Bのlowを更新してlow'と符号ビット1を出力する第1low更新部250Bと、第2rangeテーブル522Bのrange''及び再正規化の回数に基づいて再正規化を行い、第1low更新部250Bで得られたlow'を更新したlow''と符号ビット2を出力する第2low更

40

50

新部 2 5 2 B を備える。

【 0 1 0 2 】

図 3 1 も、図 2 9 と同様に処理工程とデータの流をブロック図的に表現した図であるため、実際のハードウェア的な結線とは必ずしも一致しない。例えばコンテキストメモリ部 2 1 7 B が重複して表示されているが、両者は同じものであり、図 3 1 において左側のコンテキストメモリ部 2 1 7 B はリード時、中央のコンテキストメモリ部 2 1 7 B はライト時の動作をそれぞれ示している。

(M P S E の算術符号化処理工程)

【 0 1 0 3 】

この算術符号化部 3 1 6 も、図 3 1 及び図 3 2 のフローチャートに示すようにステップ S ' 1 ~ S ' 4 の 4 段階で、シンボル A、B の算術符号化処理を行える。すなわち、ステップ S ' 1 で示す第 1 段階では、各コンテキストの読み込みが行われる。具体的には、コンテキストメモリ部 2 1 7 B で、シンボル A、B と対応するコンテキスト X、Y の読み出し (リード) を行う。

10

【 0 1 0 4 】

次にステップ S ' 2 で示す第 2 段階で、各コンテキストの更新が行われる。具体的には、コンテキストメモリ部 2 1 7 B の第 1 の出力から第 1 コンテキスト更新テーブル 2 3 0 B がコンテキスト X を読み込み、さらにシンボル A の M P S 連続数 m に基づいてコンテキスト X ' に更新する。また、更新されたコンテキスト X ' を第 2 コンテキスト更新テーブル 2 3 2 B が参照し、他方でコンテキストメモリ部 2 1 7 B の第 2 の出力から得られたコンテキスト Y、及びその M P S 連続数 n に基づき、コンテキスト Y ' に更新する。

20

【 0 1 0 5 】

さらにステップ S ' 3 で示す第 3 段階で、更新されたコンテキストの書き込みと range の更新が行われる。すなわち、第 1 コンテキスト更新テーブル 2 3 0 B 及び第 2 コンテキスト更新テーブル 2 3 2 B から各々出力されるコンテキスト X '、Y ' がコンテキストメモリ部 2 1 7 B に書き込まれる。このようにして第 1、第 2 コンテキスト更新テーブル 2 3 2 B で、M P S を処理した分のコンテキスト (発生確率) の更新をステップ S ' 2 で行い、ステップ S ' 3 でコンテキストメモリ部 2 1 7 B に書き込む。

【 0 1 0 6 】

一方でシンボル A につき、M P S 連続数 m とそのコンテキスト X に基づき、第 1 range テーブル 5 2 0 B が range の値を range ' に更新する。range の値は、range メモリ部 2 1 8 B から読み込まれる。さらにシンボル B につき、M P S 連続数 n とそのコンテキスト Y に基づき、第 2 range テーブル 5 2 2 B が range ' の値を range " に更新する。更新された range " の値は、range メモリ部 2 1 8 B に書き込まれる。

30

【 0 1 0 7 】

最後にステップ S ' 4 で示す第 4 段階で、low の更新が行われる。第 1 low 更新部 2 5 0 B は、range ' 及び再正規化の回数に応じて、low メモリ部 2 1 9 B に保持された low 値を low ' に更新すると共に、シンボル A の符号ビット 1 を出力する。一方第 2 low 更新部 2 5 2 B は、range " 及び再正規化の回数に応じて low ' を low " に更新し、low メモリ部 2 1 9 B の low 値を low " に更新する。また、シンボル B の符号ビット 2 を出力する。この方法はシンボルが M P S のときのみ有効であるが、図 2 9 と同様に簡単な演算回路で 1 サイクルで 2 シンボルの処理が可能な算術符号化部 3 1 6 が実現できる。

40

【 0 1 0 8 】

このようにして、前段のシンボルの演算を待たずに次段のシンボルの算術符号化を行うことができるので、パイプライン処理が可能となり、従来は不可能と考えられていた処理の高速化が図られる。図 3 3 に、このようなパイプライン処理の例を示す。図 3 3 に示すように、3 つのシンボルについて、算術符号化処理を並列して行うことができるので、1 サイクルで 2 シンボル以上の処理が可能となる。

50

(2つの手法の切り替え)

【 0 1 0 9 】

以上のように、2サイクル予測は1サイクル分の処理を削減し、多くの場面で予測可能である利点を有する。一方連続M P S処理予測は、数サイクル分の処理を削減でき、削減量が大きい、M P Sが連続する場合のみ予測可能であるという制約がある。

【 0 1 1 0 】

よって、これら2サイクル予測と連続M P S処理予測とを組み合わせ、状況に応じてより有効な手法を選択し、これらを切り替えて利用することで、一層の高速化を図ることができる。本発明の実施の形態3として、このような予測手法切替部を備える算術符号化装置の一例を、図23及び図24に示す。これらの図に示す算術符号化装置600も、図9と同様にコンテキスト計算部614、第1メモリ部617、第2メモリ部618、算術符号化部610を備える。これらの部材は基本的に図9等と同様であって、詳細説明を省略する。算術符号化部610は、入力された2信号をバッファリングする入力バッファ621と、2サイクル予測部622と、連続M P S処理予測部623と、これらを切り替える予測手法切替部624とを備える。予測手法切替部624は、入力されるコンテキストやrange等に応じて、その時点で適切な予測手法を選択して切り替える。図23の例では2サイクル予測部622を選択した状態を、図24の例では連続M P S処理予測部623を選択した状態を、それぞれ示している。これにより、2サイクル予測と、連続M P S処理予測を適宜組み合わせ、多くのサイクル数を削減できる。

【 0 1 1 1 】

図25に、2サイクル予測と連続M P S処理予測を予測手法切替部624で切り替える様子を示す。この図に示すように、2値化されたビット列の状態に基づいて、予測手法切替部624がこれらの切り替えを行う。ここでは、切り替えの条件として、入力ビットが"0"の連続である場合のみ連続M P S処理予測を使用し、その他の場合は2サイクル予測を使用する。予測手法切替部624はこの条件に従い、いずれかの手法を選択し、選択した手法に従って処理を行う。

(実施例)

【 0 1 1 2 】

次に本発明の実施例として、2サイクル予測、連続M P S処理予測、及びこれらを切り替える場合のそれぞれについて算術符号化を行い、削減されるサイクル数の比率をシミュレーションで求めた。ここでは、一括で入力するコンテキスト数を2として、種々のシミュレーション用画像シーケンス(akiyo、bridge-close、bridge-far、carphone、clair)に対して算術符号化を行い、そのシミュレーション結果を、それぞれ表1～表5に示す。表においてD S Eは2サイクル予測、M P S Eは連続M P S処理予測、D S E + M P S Eはこれらを切り替えた例を、それぞれ示している。この表に示すように、いずれのシーケンスに対しても2サイクル予測、連続M P S処理予測によって演算量を削減でき、特に2サイクル予測が良好な結果を示している。またこれらを切り替える方式では、一層の演算量削減が図られ、本実施の形態の有用性が確認できた。またシミュレーション結果によれば、コンテキストの変更許可数を増すほど、削減率は向上した。ただ、コンテキストの変更許可数を増加すると、削減できるサイクル数は増加するものの、回路構成が複雑化して回路面積が増大するというデメリットもあるので、サイクル数削減効果と回路構成に要するコスト、要求される仕様などに応じて適宜決定する。

【 0 1 1 3 】

10

20

30

40

【表 1】

| akiyo | | | |
|-----------|---------|--------|-------------|
| 入力コンテキスト数 | MPSE(%) | DSE(%) | MPSE+DSE(%) |
| 1 | 6.09 | | |
| 2 | 20.67 | 36.91 | 41.47 |
| 3 | 27.49 | 48.01 | 58.18 |
| 4 | 30.40 | 53.41 | 65.35 |
| 5 | 32.10 | 56.68 | 69.44 |

【 0 1 1 4 】

【表 2】

| bridge_close | | | |
|--------------|---------|--------|-------------|
| 入力コンテキスト数 | MPSE(%) | DSE(%) | MPSE+DSE(%) |
| 1 | 3.18 | | |
| 2 | 19.00 | 36.68 | 40.78 |
| 3 | 26.08 | 47.72 | 58.50 |
| 4 | 29.66 | 53.24 | 65.50 |
| 5 | 31.85 | 56.52 | 69.65 |

【 0 1 1 5 】

【表 3】

| bridge_far | | | |
|------------|---------|--------|-------------|
| 入力コンテキスト数 | MPSE(%) | DSE(%) | MPSE+DSE(%) |
| 1 | 5.60 | | |
| 2 | 20.96 | 35.72 | 41.79 |
| 3 | 27.55 | 46.40 | 58.36 |
| 4 | 31.12 | 51.81 | 65.10 |
| 5 | 33.31 | 55.00 | 69.01 |

【 0 1 1 6 】

【表 4】

| carphone | | | |
|-----------|---------|--------|-------------|
| 入力コンテキスト数 | MPSE(%) | DSE(%) | MPSE+DSE(%) |
| 1 | 5.60 | | |
| 2 | 21.50 | 37.94 | 40.85 |
| 3 | 28.30 | 49.27 | 57.97 |
| 4 | 31.73 | 54.82 | 65.34 |
| 5 | 33.65 | 58.15 | 69.75 |

【 0 1 1 7 】

【表 5】

| clair | | | |
|-----------|---------|--------|-------------|
| 入力コンテキスト数 | MPSE(%) | DSE(%) | MPSE+DSE(%) |
| 1 | 7.09 | | |
| 2 | 20.52 | 36.11 | 40.88 |
| 3 | 27.16 | 46.82 | 57.59 |
| 4 | 30.36 | 52.20 | 64.72 |
| 5 | 32.38 | 55.36 | 68.74 |

【 0 1 1 8 】

なお他の実施の形態に係る算術符号化装置は、多値信号を2値化するための2値化部と、2値化部で生成された2値シンボルを算術符号化するための算術符号化部と、コンテキスト番号を生成するためのコンテキスト計算部と、2値化部で生成された2値シンボルに

10

20

30

40

50

基づいて、優勢シンボル或いは劣勢シンボルが示すシンボル値とシンボル出現確率とを更新するための更新部と、コンテキスト計算部で計算されたコンテキスト番号をインデックスとして、シンボル値とシンボル出現確率とを格納するための確率状態格納部とを備える算術符号化装置であって、算術符号化部は、劣勢シンボルの区間を発生確率に基づいて分割した領域 $r L P S$ の予測値を格納する第 1 格納部と、同じく $r L P S$ の予測値を格納する、第 1 格納部とは別個の第 2 格納部と、第 1 格納部に格納された $r L P S$ の予測値及び初期の区間を参照して算術符号化を行い区間を更新する第 1 演算部と、同じく第 2 格納部に格納された $r L P S$ の予測値及び前記第 1 演算部で更新された更新区間を参照して算術符号化を行う、第 1 演算部とは別個の第 2 演算部と、シンボル出現確率及び優勢シンボルが示すシンボル値を更新する第 1 コンテキスト更新テーブルと、同じくシンボル出現確率及び優勢シンボルが示すシンボル値を更新する、第 1 コンテキスト更新テーブルとは別個の第 2 コンテキスト更新テーブルとを備え、第 1 コンテキスト更新テーブルは、入力されたコンテキストに対する出力を同サイクル中に前記第 2 コンテキストテーブルに入力可能な組み合わせ回路で構成されている。これにより、1 サイクル中に 2 ビット以上の処理が可能な高速な算術符号化を実現できる。

10

【0119】

算術符号化装置においては、第 1 コンテキスト更新テーブル及び第 2 コンテキスト更新テーブルを組み合わせ回路で構成できる。これによって処理の高速化が図られる。なお、第 1 コンテキスト更新テーブルを ROM で、第 2 コンテキスト更新テーブルを組み合わせ回路で構成してもよく、これによって算術符号化を安価に実現できる。

20

【産業上の利用可能性】

【0120】

本発明の算術符号化装置、算術符号化方法、算術符号化プログラム及びプログラムを格納したコンピュータで読み取り可能な記録媒体は、動画像信号を圧縮して記録したり伝送する用途に有用であり、例えば携帯端末装置、動画像撮像装置、動画像記録・再生装置等、高画質動画像を処理する装置及びその応用分野において好適に利用できる。

【図面の簡単な説明】

【0121】

【図 1】 $H.264/AVC$ の符号化処理により、画像データからビットストリームを得る動画像符号化/復号装置を示すブロック図である。

30

【図 2】 算術符号化・復号化の概念を説明するための概念図である。

【図 3】 2 値算術符号化を行う手順を示す概念図である。

【図 4】 $CABAC$ 処理を行う算術符号化装置を示すブロック図である。

【図 5】 図 4 の 2 値算術符号化部の詳細を示すブロック図である。

【図 6】 再正規化の手順を示す概念図である。

【図 7】 算術符号化装置を示すブロック図である。

【図 8】 図 7 の算術符号化装置の動作を示す概念図である。

【図 9】 本実施の形態 1 に係る算術符号化装置を示すブロック図である。

【図 10】 図 9 の第 1 メモリ部を示す概念図である。

【図 11】 本実施の形態 1 に係る算術符号化装置の変形例を示すブロック図である。

40

【図 12】 図 9 に示す算術符号化装置の算術符号化部のブロック図である。

【図 13】 図 12 の算術符号化部の動作を示す概念図である。

【図 14】 他の構成に係る算術符号化部を示すブロック図である。

【図 15】 連続 MPS 処理予測の手順を示す概念図である。

【図 16】 コンテキストが一定の場合の MPS 連続数テーブルを示す概念図である。

【図 17】 図 17 (a) 再正規化を行う様子を示す概念図、図 17 (b) は再正規化を考慮した $range$ 予測を示す概念図である。

【図 18】 再正規化を考慮した連続 MPS 処理予測部のブロック図である。

【図 19】 連続する MPS 処理において、コンテキストが変化する様子を示す概念図である。

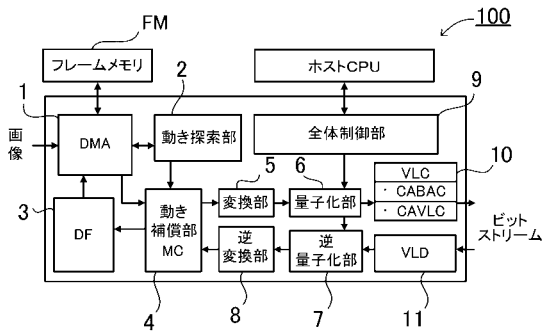
50

- 【図 2 0】M P S 連続数テーブルの例を示す概念図である。
- 【図 2 1】コンテキストが変化する場合に連続 M P S 処理予測を実現する連続 M P S 処理予測部を構成するブロック図である。
- 【図 2 2】詳細な連続 M P S 処理予測部のブロック図を示す。
- 【図 2 3】本発明の実施の形態 3 に係る算術符号化装置を示すブロック図である。
- 【図 2 4】本発明の実施の形態 3 に係る算術符号化装置を示すブロック図である。
- 【図 2 5】2 サイクル予測と連続 M P S 処理予測を予測手法切替部で切り替える様子を示すブロック図である。
- 【図 2 6】H . 2 6 4 / A V C による算術符号化・復号の処理プロセスのフローチャートである。 10
- 【図 2 7】H . 2 6 4 / A V C の算術符号化の処理プロセスを具現化した算術符号化装置を示すブロック図である。
- 【図 2 8】従来の算術符号化装置を示すブロック図である。
- 【図 2 9】2 サイクル予測を行う算術符号化部の変形例を示すブロック図である。
- 【図 3 0】図 2 9 の算術符号化部による算術符号化の手順を示すフローチャートである。
- 【図 3 1】連続 M P S 処理予測を行う算術符号化部の変形例を示すブロック図である。
- 【図 3 2】図 3 1 の算術符号化部による算術符号化の手順を示すフローチャートである。
- 【図 3 3】図 2 9、図 3 1 の算術符号をパイプライン処理で行う様子を示すタイミングチャートである。 20
- 【符号の説明】 20
- 【 0 1 2 2 】
- 1 ... D M A
- 2 ... 動き探索部
- 3 ... デブロッキングフィルタ
- 4 ... 動き補償部
- 5 ... 変換部
- 6 ... 量子化部
- 7 ... 逆量子化部
- 8 ... 逆変換部
- 9 ... 全体制御部 30
- 1 0 ... 算術符号化 / 復号装置
- 1 0 A ... 算術符号化装置
- 1 1 ... 可変長復号化部
- 1 2、1 1 2 ... 2 値化部
- 1 4、1 1 4、6 1 4 ... コンテキスト計算部
- 1 6、1 1 6 ... 2 値算術符号化部
- 2 0 ... r L P S テーブル
- 3 0 ... コンテキスト変更テーブル
- 4 0、1 4 0 ... 演算部 40
- 5 0 ... 再正規化部
- 7 0、1 7 0 ... コンテキスト更新部
- 1 0 0 ... 動画像符号化 / 復号装置
- 1 1 7 ... 第 1 メモリ部
- 1 1 7 A ... メモリ部
- 1 1 8 ... 第 2 メモリ部
- 1 1 9 ... 予測部
- 1 2 0 ... 第 1 r L P S テーブル
- 1 2 1 ... 出力バッファ
- 1 2 2 ... 第 2 r L P S テーブル
- 1 6 0、3 6 0 ... 符号ビット生成部 50

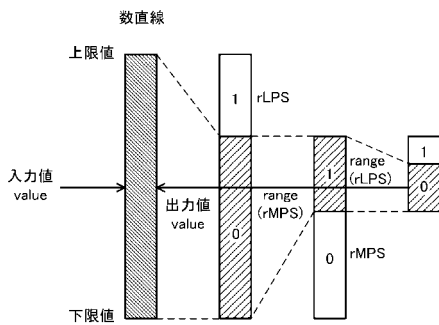
| | |
|--|----|
| 2 0 0、3 0 0 ... 算術符号化装置 | |
| 2 1 6、3 1 6 ... 算術符号化部 | |
| 2 1 7、2 1 7 B ... コンテキストメモリ部 | |
| 2 1 8、2 1 8 B ... r a n g eメモリ部 | |
| 2 1 9、2 1 9 B ... l o wメモリ部 | |
| 2 2 0 ... 第 1 r L P Sテーブル | |
| 2 2 2 ... 第 2 r L P Sテーブル | |
| 2 2 4 ... 第 1 選択器 | |
| 2 2 6 ... 第 2 選択器 | |
| 2 3 0、2 3 0 B ... 第 1 コンテキスト更新テーブル | 10 |
| 2 3 2、2 3 2 B ... 第 2 コンテキスト更新テーブル | |
| 2 4 0 ... 第 1 r a n g e更新部 | |
| 2 4 2 ... 第 2 r a n g e更新部 | |
| 2 5 0、2 5 0 B ... 第 1 l o w更新部 | |
| 2 5 2、2 5 2 B ... 第 2 l o w更新部 | |
| 3 0 1 ... 第 1 算術符号化手段 | |
| 3 0 2 ... 第 2 算術符号化手段 | |
| 3 2 0 ... 第 1 r L P Sテーブル | |
| 3 2 2 ... 第 2 r L P Sテーブル | |
| 3 3 0 ... 第 1 コンテキスト更新テーブル | 20 |
| 3 3 2 ... 第 2 コンテキスト更新テーブル | |
| 3 4 0 ... 第 1 演算部 | |
| 3 4 2 ... 第 2 演算部 | |
| 3 5 0 ... 第 1 再正規化部 | |
| 3 5 2 ... 第 2 再正規化部 | |
| 4 0 0、4 0 0 B、5 0 0 ... 連続 M P S 処理予測部 | |
| 4 2 0 ... r a n g eテーブル | |
| 4 2 0 A、4 2 0 B ... 予測テーブル | |
| 4 3 0 ... コンテキスト更新テーブル | |
| 4 5 0 ... 再正規化部 | 30 |
| 5 2 0、5 2 0 B ... 第 1 r a n g eテーブル | |
| 5 2 2、5 2 2 B ... 第 2 r a n g eテーブル | |
| 5 3 0 ... 第 1 コンテキスト更新テーブル | |
| 5 3 2 ... 第 2 コンテキスト更新テーブル | |
| 5 5 0 ... 再正規化部 | |
| 6 0 0 ... 算術符号化装置 | |
| 6 1 0 ... 算術符号化部 | |
| 6 1 7 ... 第 1 メモリ部 | |
| 6 1 8 ... 第 2 メモリ部 | |
| 6 2 1 ... 入力バッファ | 40 |
| 6 2 2 ... 2 サイクル予測部 | |
| 6 2 3 ... 連続 M P S 処理予測部 | |
| 6 2 4 ... 予測手法切替部 | |
| 7 0 0 ... 算術符号化装置 | |
| 7 1 0 ... 適応算術符号化・復号部 | |
| 7 1 1 ... 算術符号化・復号部 | |
| 7 1 2 ... シンボル出現確率制御部 | |
| 7 1 3 ... 確率状態格納部 | |
| 7 2 0 ... コンテキスト計算部 | |
| 7 3 0 ... 符号化・復号制御部 | 50 |

- 8 0 0 ... 算術符号化装置
- 8 1 0 ... 適応算術符号化部
- 8 1 1 ... 算術符号化部
- 8 1 4 ... シンボル出現確率制御部
- 8 1 5 ... 確率状態格納部
- 8 2 0 ... コンテキスト計算部
- 8 2 1 ... レジスタ
- 8 2 2 ... 比較判定部
- 8 3 0 ... 符号化制御部
- F M ... フレームメモリ

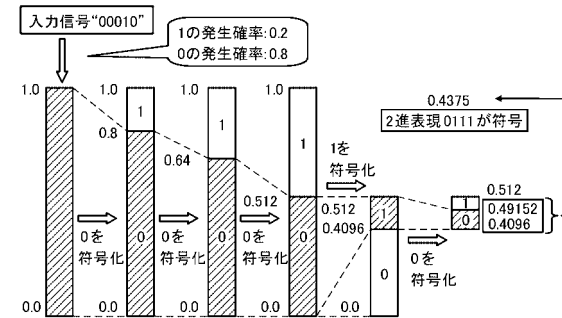
【図1】



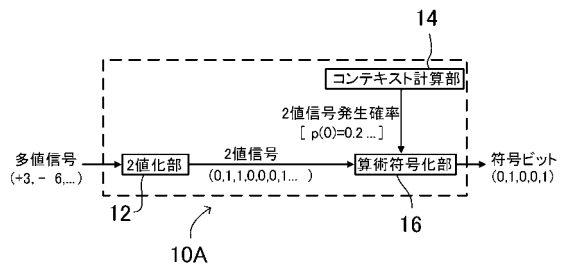
【図2】



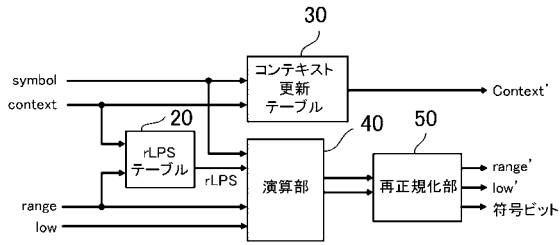
【図3】



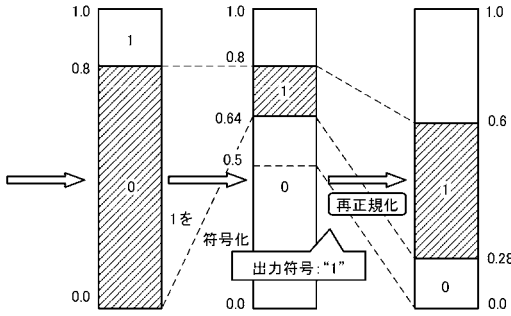
【図4】



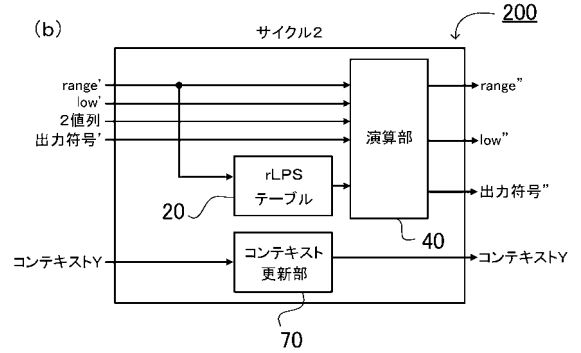
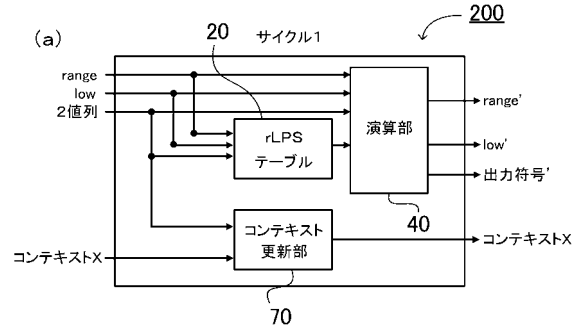
【図5】



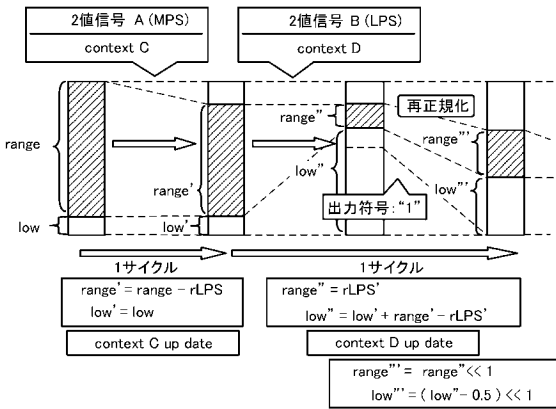
【図6】



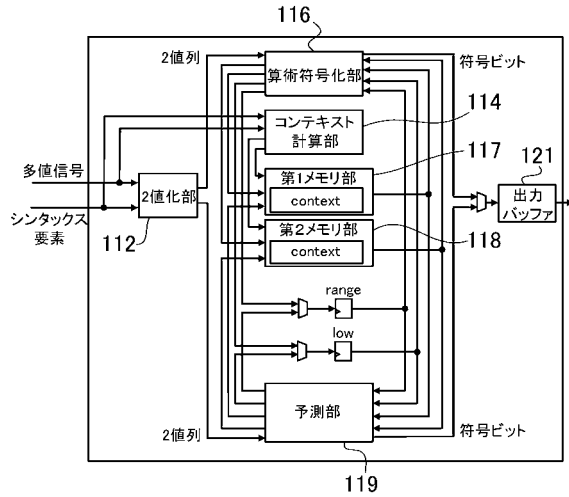
【図7】



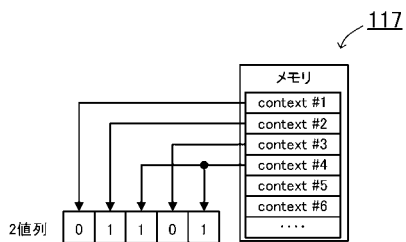
【図8】



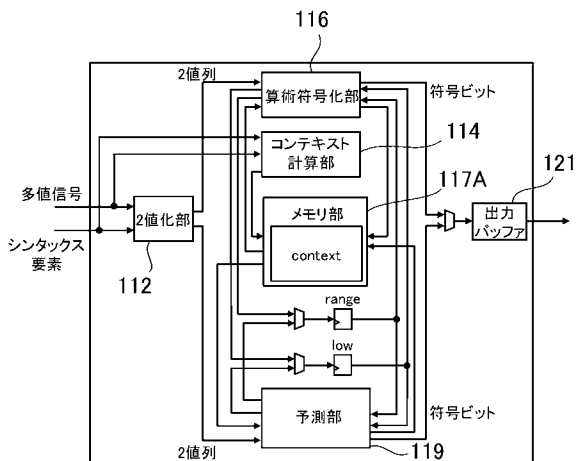
【図9】



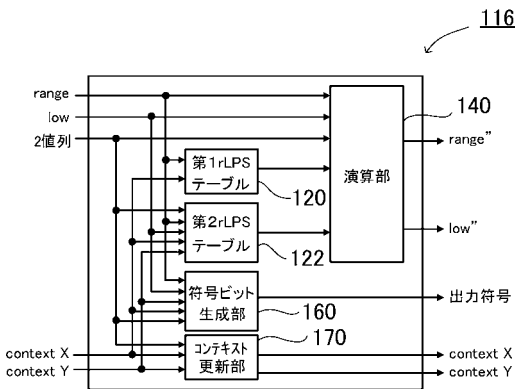
【図10】



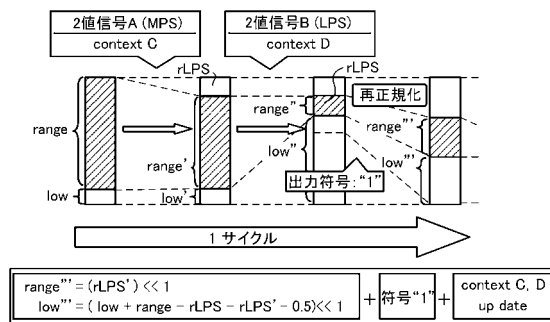
【図11】



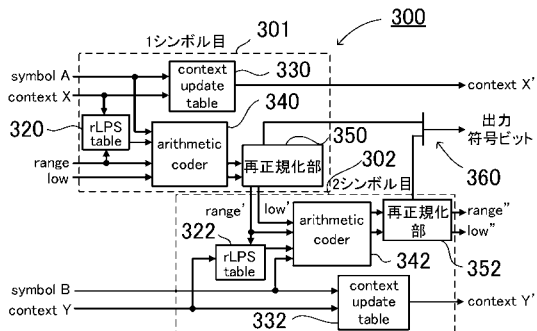
【図12】



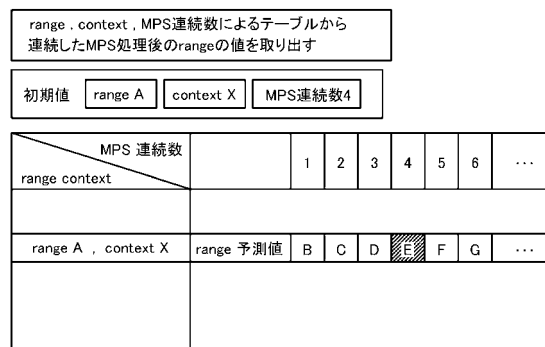
【図13】



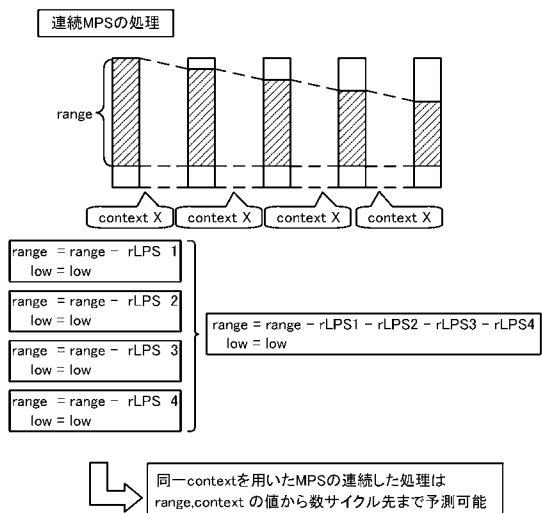
【図14】



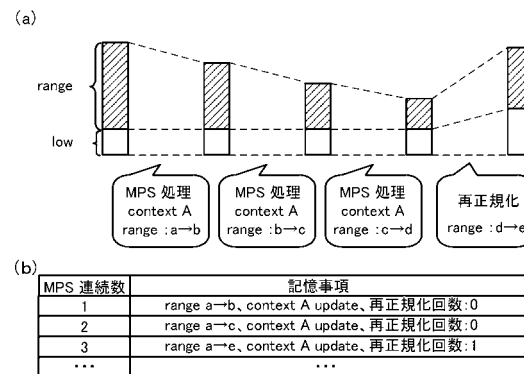
【図16】



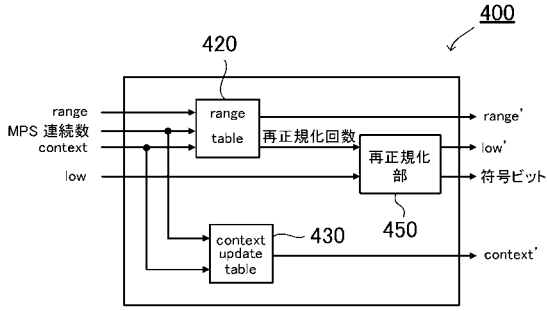
【図15】



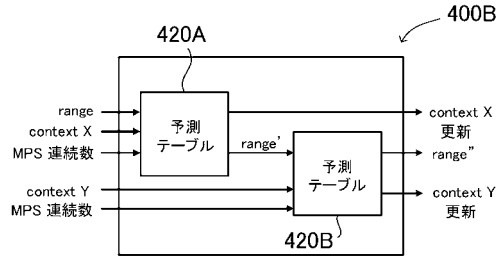
【図17】



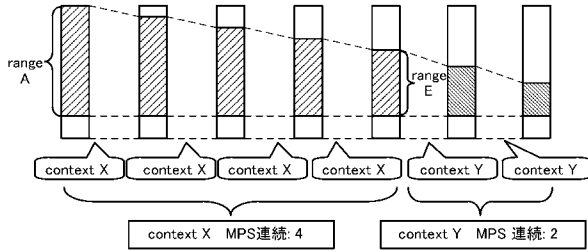
【図18】



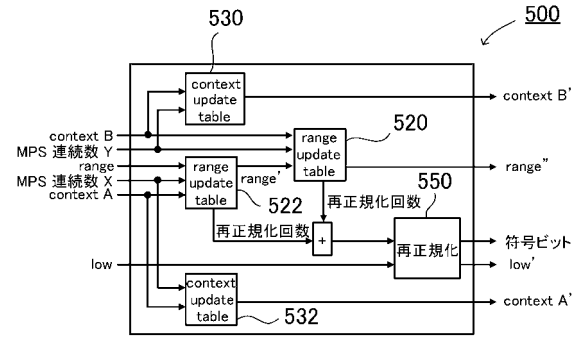
【図21】



【図19】



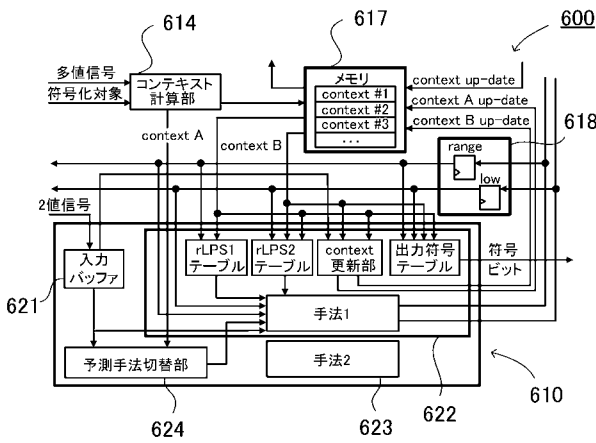
【図22】



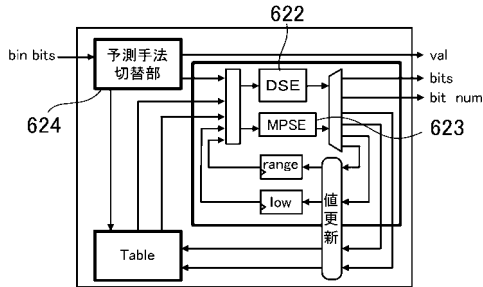
【図20】

| MPS 連続数 | 1 | 2 | 3 | 4 | 5 | 6 | ... |
|--------------------|-----------|---|---|---|---|---|-----|
| range context | | | | | | | |
| range A, context X | range 予測値 | B | C | D | E | F | G |
| ... | | | | | | | |
| range E, context Y | range 予測値 | H | I | J | K | L | M |

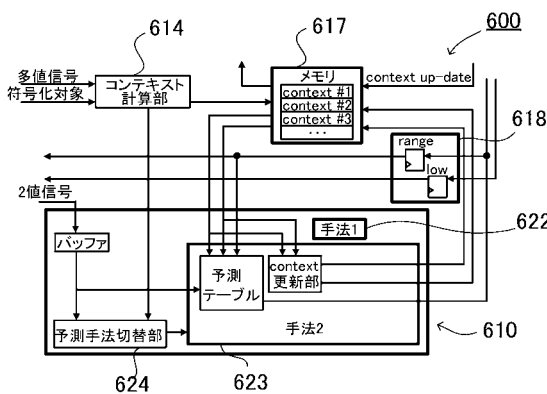
【図23】



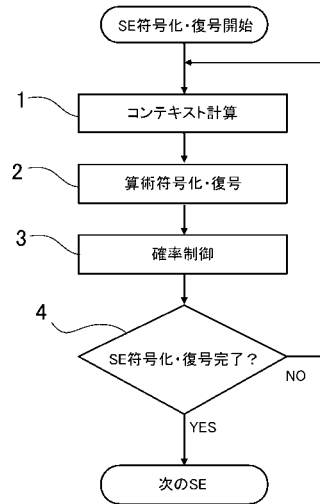
【図25】



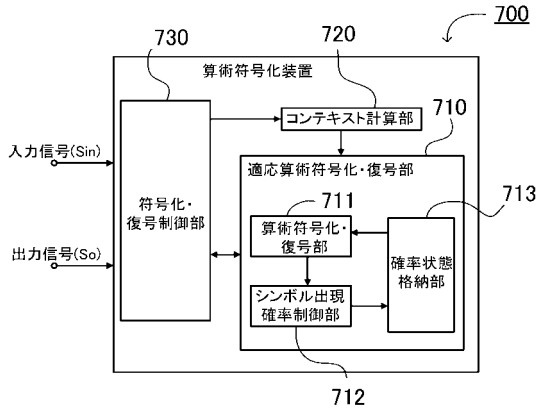
【図24】



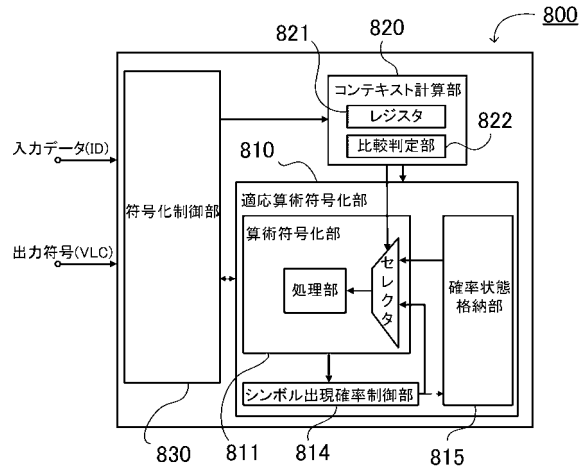
【図26】



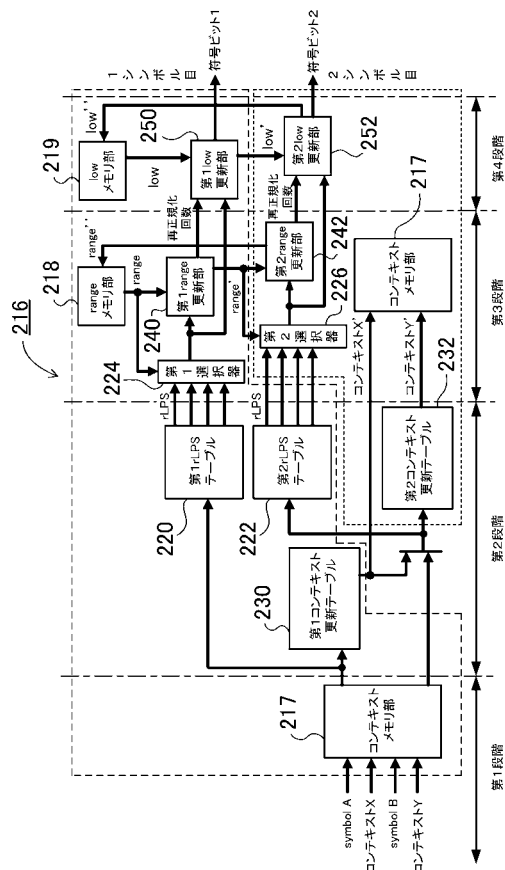
【図27】



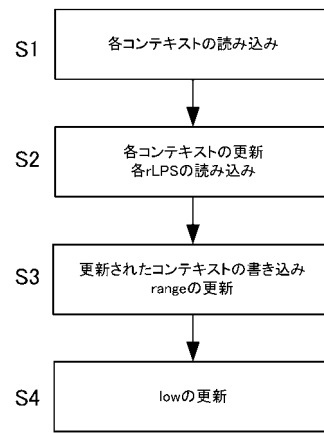
【図28】



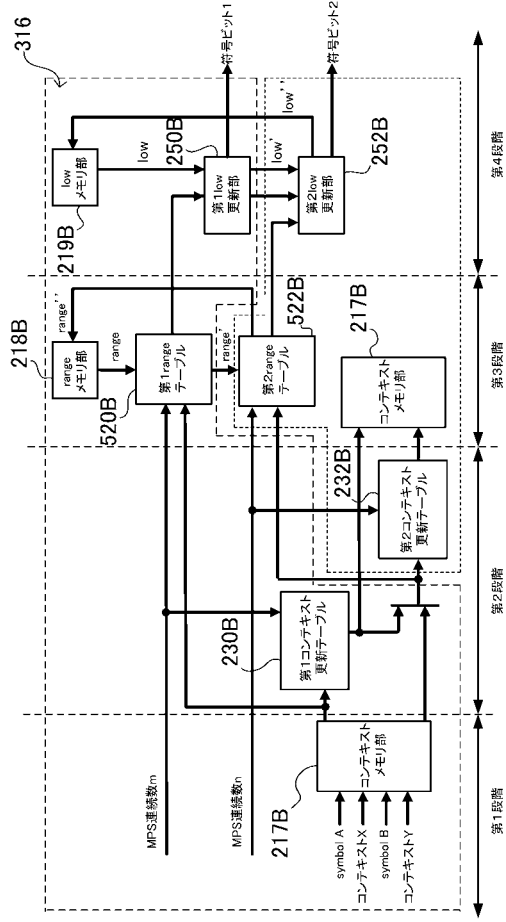
【図29】



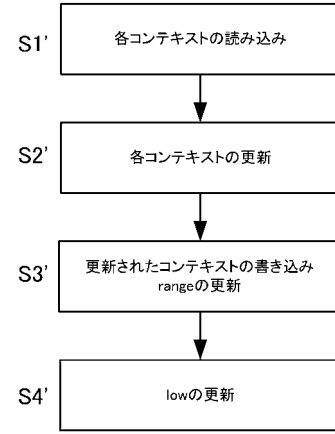
【図30】



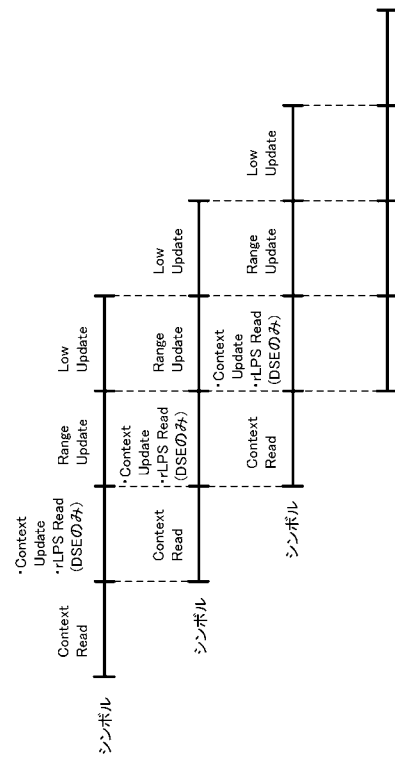
【 図 3 1 】



【 図 3 2 】



【 図 3 3 】



フロントページの続き

(72)発明者 島本 隆

徳島県徳島市南常三島町2丁目1番地 国立大学法人徳島大学大学院ソシオテクノサイエンス研究
部内

審査官 渡辺 未央子

(56)参考文献 特開2005-130099(JP,A)

特開2005-504471(JP,A)

特開2005-525018(JP,A)

山田篤, 他, H.264/AVCにおけるCABAC高速処理アーキテクチャ, 電子情報通信学会総合大会講演
論文集, 2006年 3月 8日, D-11-39, p.39

Tian Song, et al., Efficient Multistate Estimated Architecture of CABAC for H.264/AVC
Encoder, Journal of Signal Processing, 2008年 1月, Vol.12, No.1, pp.65-71

(58)調査した分野(Int.Cl., DB名)

H03M 7/40

H04N 7/26