

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3684128号
(P3684128)

(45) 発行日 平成17年8月17日(2005.8.17)

(24) 登録日 平成17年6月3日(2005.6.3)

(51) Int. Cl.⁷

F I

HO4N	1/41	HO4N	1/41	B
HO3M	7/40	HO3M	7/40	
HO4N	1/413	HO4N	1/413	Z
HO4N	7/24	HO4N	7/13	Z

請求項の数 12 (全 23 頁)

(21) 出願番号	特願2000-41693 (P2000-41693)	(73) 特許権者	000001007
(22) 出願日	平成12年2月18日(2000.2.18)		キヤノン株式会社
(65) 公開番号	特開2001-230935 (P2001-230935A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成13年8月24日(2001.8.24)	(74) 代理人	100076428
審査請求日	平成14年12月26日(2002.12.26)		弁理士 大塚 康德
		(74) 代理人	100101306
			弁理士 丸山 幸雄
		(74) 代理人	100115071
			弁理士 大塚 康弘
		(72) 発明者	中山 忠義
			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
		(72) 発明者	神戸川 実
			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】 算術符号化／復号化方法ならびに算術符号化／復号化装置

(57) 【特許請求の範囲】

【請求項1】

1 画素をmビット(m 2)で表される多値画像データを、各画素を構成する1ビット毎に算術符号化する算術符号化方法であって、

符号化対象画素を構成する各符号化対象ビットを符号化する為に、2サイクル以上前に符号化されたビットを用いてコンテキスト情報を生成するコンテキスト生成ステップと、

前記符号化対象ビットを、前記コンテキスト情報に基づいて算術符号化する算術符号化ステップとを備え、

前記多値画像データを構成するビットプレーンの数をm、該ビットプレーン番号をMSBから順に1、2、・・・mとし、各画素の番号を{ }、ビットプレーンを[]で表すとすると、前記算術符号化するビットの順序として、{ i } [j]の次は、j mの場合{ i 1 } [j + 1]を、j = mの場合{ i + m } [1]を算術符号化することを特徴とする算術符号化方法。

【請求項2】

符号化する画素の数がnである場合、{ i 1 } 0及び{ i + 1 } > nでは、所定値を符号化することを特徴とする請求項1に記載の算術符号化方法。

【請求項3】

符号化する画素の数がnである場合、{ i 1 } 0及び{ i + 1 } > nでは、符号化をスキップすることを特徴とする請求項1に記載の算術符号化方法。

【請求項4】

10

20

1画素を m ビット($m \geq 2$)で表される多値画像データを、各画素を構成する1ビット毎に算術復号化する算術復号化方法であって、

復号化対象画素を構成する各復号化対象ビットを復号化する為に、2サイクル以上前に復号化されたビットを用いてコンテキスト情報を生成するコンテキスト生成ステップと、

前記復号化対象ビットを、前記コンテキスト情報に基づいて算術復号化する算術復号化ステップとを備え、

前記多値画像データを構成するビットプレーンの数を m 、該ビットプレーン番号をMSBから順に1、2、 \dots 、 m とし、各画素の番号を $\{ \}$ 、ビットプレーンを $[]$ で表すとすると、前記算術復号化するビットの順序として、 $\{ i \} [j]$ の次は、 $j = m$ の場合 $\{ i - 1 \} [j + 1]$ を、 $j = 1$ の場合 $\{ i + m \} [1]$ を算術復号化することを特徴とする算術復号化方法。

【請求項5】

復号化する画素の数が n である場合、 $\{ i - 1 \} = 0$ 及び $\{ i + 1 \} > n$ では、所定値を復号化することを特徴とする請求項4に記載の算術復号化方法。

【請求項6】

復号化する画素の数が n である場合、 $\{ i - 1 \} = 0$ 及び $\{ i + 1 \} > n$ では、復号化をスキップすることを特徴とする請求項4に記載の算術復号化方法。

【請求項7】

1画素を m ビット($m \geq 2$)で表される多値画像データを、各画素を構成する1ビット毎に算術符号化する算術符号化装置であって、

符号化対象画素を構成する各符号化対象ビットを符号化する為に、2サイクル以上前に符号化されたビットを用いてコンテキスト情報を生成するコンテキスト生成手段と、

前記符号化対象ビットを、前記コンテキスト情報に基づいて算術符号化する算術符号化手段とを備え、

前記多値画像データを構成するビットプレーンの数を m 、該ビットプレーン番号をMSBから順に1、2、 \dots 、 m とし、各画素の番号を $\{ \}$ 、ビットプレーンを $[]$ で表すとすると、前記算術符号化するビットの順序として、 $\{ i \} [j]$ の次は、 $j = m$ の場合 $\{ i - 1 \} [j + 1]$ を、 $j = 1$ の場合 $\{ i + m \} [1]$ を算術符号化することを特徴とする算術符号化装置。

【請求項8】

1画素を m ビット($m \geq 2$)で表される多値画像データを、各画素を構成する1ビット毎に算術復号化する算術復号化装置であって、

復号化対象画素を構成する各復号化対象ビットを復号化する為に、2サイクル以上前に復号化されたビットを用いてコンテキスト情報を生成するコンテキスト生成手段と、

前記復号化対象ビットを、前記コンテキスト情報に基づいて算術復号化する算術復号化手段とを備え、

前記多値画像データを構成するビットプレーンの数を m 、該ビットプレーン番号をMSBから順に1、2、 \dots 、 m とし、各画素の番号を $\{ \}$ 、ビットプレーンを $[]$ で表すとすると、前記算術復号化するビットの順序として、 $\{ i \} [j]$ の次は、 $j = m$ の場合 $\{ i - 1 \} [j + 1]$ を、 $j = 1$ の場合 $\{ i + m \} [1]$ を算術復号化することを特徴とする算術復号化装置。

【請求項9】

予測値及び確率推定値を保持する、各ビットプレーン毎に設けられた確率推定値メモリと、

前記生成されたコンテキスト情報に基づいて予測値及び確率推定値を更新する予測推定更新手段とを有し、

前記生成されたコンテキスト情報をアドレスとして、前記確率推定値メモリから予測値及び確率推定値を読み出すことを特徴とする請求項8に記載の算術復号化装置。

【請求項10】

予測値及び状態値を保持する、各ビットプレーン毎に設けられた予測状態メモリと、

10

20

30

40

50

前記生成されたコンテキスト情報に基づいて予測値及び状態値を更新する予測状態更新手段とを有し、

前記生成されたコンテキスト情報をアドレスとして、前記予測状態メモリから予測値及び状態値を読み出すことを特徴とする請求項 8 に記載の算術復号化装置。

【請求項 11】

復号化する画素の数が n である場合、 $\{i-1\} = 0$ 及び $\{i+1\} > n$ では、所定値を復号化することを特徴とする請求項 8 に記載の算術復号化装置。

【請求項 12】

復号化する画素の数が n である場合、 $\{i-1\} = 0$ 及び $\{i+1\} > n$ では、復号化をスキップすることを特徴とする請求項 8 に記載の算術復号化装置。

10

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、算術符号 (Arithmetic code) を用いて画像データを符号化する算術符号化方法と、該方法で符号化された画像データを復号化する算術復号化装置に関し、特に多値画像の算術符号化方法と算術復号化装置に関する。

【0002】

【従来の技術】

算術符号は、 $[0, 1)$ の数値直線上の対応区間 (2進小数で $[0.0 \dots 0, 0.1 \dots 1]$) を各シンボルの生起確率に応じて不等長に分割していき、符号化対象シンボル系列を対応する部分区間に割り当て、再帰的に分割を繰り返していくことにより得られた区間内に含まれる点の座標を少なくとも他の区間と区別できる 2進小数で表現して、そのまま符号とするものである。

20

【0003】

算術符号の方法の代表的なものとして、ITU の機関である JBIG (Joint Bi-level Image Experts Group) で標準化された JBIG (QM-coder) 方法、IBM 社が提案している Q-coder 等がある。

【0004】

各方法により使用されている用語が異なるため、ここでは標準方法である JBIG 方法に用いられている用語を使用するが、これにより JBIG 方法の算術符号に限定されるものではない。

30

【0005】

算術符号化では、原理的には算術演算部で乗算処理を必要とするが、乗算処理を行うと算術演算部のハードウェア規模が大きくなったり、乗算処理にかかる処理時間が長くなるといった問題があるため、該算術演算を加減演算に置き換え簡略化した方法が主流になっている。

【0006】

算術演算部内部には、符号化シンボル系列に対応する領域幅 (オージェント) を保持する A レジスタと、コードを発生する基となる値を保持するコードレジスタ (C レジスタ) とがあり、符号化予測値に対する符号化対象シンボルのはずれ確率を確率推定値 LSZ として入力し、前記シンボルが予測値に一致したかあるいは不一致かという情報に基づいて、確率推定値 LSZ と前記 2 つのレジスタの値から算術演算 (加減算) を行ない、該 2 つのレジスタの値を更新する。

40

【0007】

確率推定値 LSZ は、符号化対象シンボルの周辺シンボル情報を数値化したコンテキストごとに持つ $index$ (状態) 値から所定の対応表を用いて変換することによって得られる。コンテキストごとに持つ状態値は、所定条件に基づいて更新することにより、符号化シンボル系列固有のパターンを学習することができ、符号化効率を上げることができる。

【0008】

JBIG 方法では、A レジスタの値を $10000H$ に初期化し、算術演算 (符号化あるいは

50

は復号化)の結果、Aレジスタの値が8000H未満になると、正規化処理と称して、Aレジスタの値が8000H以上になるまで左シフトを行い、その際に現コンテキストの状態値を更新する。

【0009】

また、Aレジスタの値を左シフトする時、それと同時にCレジスタの値も左シフトする。この時、Cレジスタの再上位からシフトアウトされるデータが符号化データとなる。

【0010】

符号化対象のシンボルの値が0または1しかとらない、2値の画素データである場合、上記算術符号化処理を行なう符号化ブロックの構成は、図1に示すものとなる。

【0011】

<符号化ブロック>

同図において、101は符号化対象画素の2値データPIXを入力する端子、102は2値データPIX信号、103は符号化対象画素の周辺画素データからなるコンテキスト情報、105はコンテキスト情報ごとに予測値とindex(状態)値を保持する予測状態メモリ、106は予測状態メモリ105から出力される予測値、107は予測状態メモリ105から出力される状態値、109は出力された状態値107を確率推定値LSZに変換する確率推定部、111は確率推定部109から出力される確率推定値LSZ、113は排他的NORゲート、114は前記2値データPIXと予測値106との一致/不一致を表わす信号、115は算術符号化演算部、117は算術符号化演算部115から出力される符号化データ(コードデータ)、118は前記予測状態メモリ105の更新を要求するための信号、119は現在の予測値106と状態値107から更新後の予測値と状態値を求める予測状態更新部、120は予測状態更新部119から出力される、更新後の予測値・状態値信号および予測状態メモリ105に該信号を書き込む制御信号等である。

【0012】

次に、図1に示す算術符号化処理ブロックの動作説明を行なう。

【0013】

符号化対象画素の2値データPIXは端子101から入力され、排他的NORゲート113に入力する。一方、コンテキスト情報である103の符号化対象画素の周辺画素データCXi(i=1,2,...,n)が予測状態メモリ105に与えられる。

【0014】

予測状態メモリ105は周辺画素データからなるコンテキスト情報103をアドレスとして、符号化対象画素に対する予測値106とindex(状態)値107を出力する。なお、この状態値107はそれまでに符号化処理して学習した内容に基づいて決まる。予測値106は、0または1の2値であり、排他的NORゲート113に入力される。

【0015】

一方、状態値107は確率推定部109に送られ、ここで確率推定値111に変換される。確率推定値111は、2値データPIXが予測値106と一致しない確率を表わす。状態値107から確率推定値111への変換は、ROMテーブルあるいはデコーダを用いて行なわれる。また、状態値107は予測状態更新部119にも入力される。

【0016】

排他的NORゲート113では、2値データPIXと予測値106との一致/不一致の関係を調べ、一致していれば1を、不一致であれば0を算術符号化演算部115へ送る。

【0017】

算術符号化演算部115では、一致/不一致情報114と、確率推定部109から入力される確率推定値111とに基づいて、後述する符号化処理のための算術演算を行ない、符号化データ117を出力する。

【0018】

算術符号化演算部115における算術演算の過程で上述の正規化処理が行なわれると、信号118を出力し、予測状態更新部119に更新要求を行なう。

【0019】

10

20

30

40

50

予測状態更新部 119 は、入力される状態値 107、予測値 106 及び一致 / 不一致情報 114 から、更新後の状態値と予測値を求めて、更新データを予測状態メモリ 105 に送る。更新データを受信した予測状態メモリ 105 は、更新要求に基づいて保持内容を更新する。なお、予測状態更新部 119 において入力状態値 106 から更新後の状態値を求めるには、ROM テーブルあるいはデコーダ等を用いる。

【0020】

次に、図 2 に算術符号化演算部 115 の一構成例を示し、その動作について述べる。

【0021】

同図において、201 は領域幅 (オージェント) を保持する A レジスタ、203 はコードを発生する基となる値を保持するコードレジスタ (C レジスタ)、205 は A レジスタ 201 の出力値から確率推定値 111 を減算する減算器、207 は C レジスタ 203 の出力値に減算器 205 の出力を加算する加算器、209 は A レジスタ 201 へ入力するデータを選択するセレクタ、211 は C レジスタへ入力するデータを選択するセレクタ、213 は C レジスタからシフトアウトされるコード情報を取り込むシフトレジスタ、215 は出力レジスタ、217 はシフトレジスタ 213 に取り込んだビット数をカウントするカウンタ、218 はシフトクロックを入力する端子、219 はマスク回路、220 はマスク回路 219 によりマスクされたシフトクロック、221 は、A レジスタ 201 の出力値が 8000H (16 進) 以上であることを検出する検出器である。

10

【0022】

初期化後の状態では、A レジスタ 201 の値は 10000H、C レジスタ 203 の値は 0H であり、カウンタ 217 の値は 0 である。

20

【0023】

入力された一致 / 不一致情報 114 と確率推定値 111 に基づいて所定の算術演算を行うことにより、1 画素を符号化するごとに A レジスタ 201 と C レジスタ 203 の値を更新する。そして、A レジスタ 201 の正規化処理に伴い C レジスタ 203 の上位からシフトアウトされるデータを N ビット単位にまとめて出力したものが符号化データ 117 となる。

【0024】

まず入力された確率推定値 111 は減算器 205 とセレクタ 209 に入力する。

【0025】

減算器 205 では、A レジスタ 201 の出力値から、確率推定値 111 を減算し、その結果をセレクタ 209 と加算器 207 へ送る。

30

【0026】

セレクタ 209 には確率推定値 111 も与えられており、一致 / 不一致情報 114 が 1 の時は減算結果を、0 の時は確率推定値 111 を選択し出力する。

【0027】

加算器 207 では、減算器 205 から送られてきた減算結果を C レジスタ 203 の出力値に加算し、加算結果をセレクタ 211 に送る。セレクタ 211 には、C レジスタ 203 の出力値も直接与えられており、前記一致 / 不一致情報 114 が 1 の時は C レジスタ 203 の出力値を、0 の時は加算結果を選択し出力する。

40

【0028】

セレクタ 209、211 それぞれの出力は、それぞれ次のサイクルで、A レジスタ 201、C レジスタ 203 に取り込まれる。

【0029】

A レジスタ 201 の値が 8000H 未満の時は、検出器 221 でそれを検出し、次の算術演算を行う前に正規化処理を行う。それに対し、A レジスタ 201 の値が 8000H 以上の時は、次の算術演算を行う。

【0030】

正規化処理の 1 つの実現方法は、A レジスタ 201 と C レジスタ 203 のそれぞれにシフト機能を有するレジスタを用い、該レジスタにシフト用のクロック 220 を与え、A レジ

50

スタ 201 の値が 8000H になるまで、AレジスタとCレジスタを左シフトするものである。

【0031】

シフト用のクロック 220 は、検出器 221 の出力に基づいて、端子 218 から入力されるシフトクロックをマスク回路 219 にてマスクすることにより得られる。

【0032】

Aレジスタ 201 の値が 8000H 未満のあいだ、入力シフトクロックはマスク回路 219 を通過し、Aレジスタ 201、Cレジスタ 203 並びに、シフトレジスタ 213、カウンタ 217 に送られる。このシフトクロックが入力する毎に、Aレジスタ 201、Cレジスタ 203、シフトレジスタ 213 は 1 ビット左へシフトし、カウンタ 217 は 1 つカウントアップする。

10

【0033】

シフトレジスタ 213 へ入力するデータは、Cレジスタ 203 の最上位ビットであるため、シフトクロック 220 が各ブロックへ送られるたびに、Cレジスタ 203 の上位ビットデータがシフトレジスタ 213 に移ることになる。

【0034】

シフトレジスタ 213 に移されたビット数はシフトクロック 220 によりカウントアップするカウンタ 217 でカウントされていることになり、該カウント値が所定の値に達したところで、カウンタ 217 は出力レジスタ 215 へデータ取り込みパルス 222 を送る。

【0035】

出力レジスタ 215 はパルス 222 が入力されるタイミングで、シフトレジスタ 213 から送られる所定ビット数のデータを一括して受け取り、次のデータ取り込みパルス 222 が入力されるまで、該データを保持する。

20

【0036】

なお、Aレジスタ 201 の値が 8000H 未満であることを検出器 221 が検出したときに、不図示の回路により予測状態更新部 119 へ更新を要求する信号 118 が出力される。

【0037】

以上説明したような算術演算及び正規化処理により、端子 101 に入力された 2 値データ P I X が符号化され、符号化データ 117 が算術符号化演算部 115 より所定ビット単位

30

【0038】

<復号化ブロック>

次に、図 1 に示す符号化ブロックに対応した復号化ブロックの構成を図 3 に示し、その動作を説明する。

【0039】

図 3 に示す復号化ブロックと、図 1 に示す符号化ブロックとの大きな違いは、以下の 2 点である。

【0040】

(1) 算術符号化演算部 115 の代わりに、算術復号化演算部 301 を用いる。(2) 符号化データ 306 を入力して、着目画素データの値を求める。

40

その他の構成は基本的に同じであるため、上記違いについて簡単に説明する。

【0041】

まず算術復号化演算部 301 内部の動作説明から行なう。図 4 に、算術復号化演算部 301 の構成を示す。

【0042】

同図において、減算器 401 は図 2 の算術符号化演算部 115 における加算器 207 を、入力バッファレジスタ 403 は図 2 の出力レジスタ 215 を、また、シフトレジスタ 405 はシフトレジスタ 213 をそれぞれ置き換えたものである。

【0043】

50

入力された確率推定値 1 1 1 は、減算器 2 0 5 とセレクタ 2 0 9 に入力される。減算器 2 0 5 では A レジスタ 2 0 1 の出力値から、入力された確率推定値 1 1 1 を減算し、その減算結果をセレクタ 2 0 9 と減算器 4 0 1 へ送る。

【 0 0 4 4 】

図 1 の算術符号化演算部 1 1 5 では、一致 / 不一致情報 1 1 4 に基づいてセレクタ 2 0 9 が動作していたが、算術復号化演算部 3 0 1 では、減算器 4 0 1 における減算結果が正か負かによって、セレクタ 2 0 9 の動作が決まる。

【 0 0 4 5 】

減算器 4 0 1 では、C レジスタ 2 0 3 の出力値から減算器 2 0 5 の出力値を減算し、減算結果をセレクタ 2 1 1 に送り、更に減算結果が正か負かを表す信号 3 0 2 を出力する。信号 3 0 2 は、減算結果が正の場合 0、負の場合 1 という値をとる。

10

【 0 0 4 6 】

この信号 3 0 2 を制御信号として、減算結果が正の場合、セレクタ 2 0 9 は確率推定値 1 1 1 を選択し、減算結果が負の場合、減算器 2 0 5 の出力を選択して出力する。

【 0 0 4 7 】

セレクタ 2 1 1 もセレクタ 2 0 9 と同じ信号 3 0 2 により制御される。セレクタ 2 1 1 には減算器 4 0 1 の出力すなわち減算結果と、C レジスタ 2 0 3 の出力値が入力されている。そして、減算結果が正の場合、減算器 4 0 1 の出力を選択し、減算結果が負の場合、C レジスタ 2 0 3 の出力を選択して出力する。

【 0 0 4 8 】

20

セレクタ 2 0 9、2 1 1 の出力は、それぞれ次のサイクルで、A レジスタ 2 0 1、C レジスタ 2 0 3 に取り込まれ、算術符号化演算部 1 1 5 と同様に正規化処理を行う。

【 0 0 4 9 】

算術符号化演算部 1 1 5 では、正規化処理時に符号化データ 1 1 7 を出力したが、算術復号化演算部 3 0 1 では、正規化処理にともない符号化データ 3 0 6 を取り込んでゆく。

【 0 0 5 0 】

符号化データ 3 0 6 は、図 1 の算術符号化演算部 1 1 5 から出力され、不図示の記憶ユニットに一旦格納される。そしてしかるべきタイミングで読み出されて、復号化ブロック内の算術復号化演算部 3 0 1 に与えられる。あるいは、符号化データをそのまま遠隔地まで転送し、遠隔地にて復号化ブロックを用いて、元の 2 値画像データに復元される。復元された 2 値画像データは、モニタやプリンタ等の画像表示機器に送られ、可視画像に変換される。

30

【 0 0 5 1 】

算術復号化演算部 3 0 1 に与えられた符号化データ 3 0 6 は、入力バッファレジスタ 4 0 3 を経由してシフトレジスタ 4 0 5 に取り込まれ、正規化処理によって、シフトレジスタ 4 0 5 の最上位ビットが C レジスタ 2 0 3 の最下位ビットにシフト入力される。正規化処理の制御方法は算術符号化演算部 1 1 5 と同じであるため、その説明は省略する。なお、この正規化処理が行われるタイミングで、不図示の回路により予測状態更新部 1 1 9 へ更新を要求する信号 3 1 5 が出力される。

【 0 0 5 2 】

40

以上説明した算術復号化演算部 3 0 1 の演算処理により、復号中の着目画素値が予測値に一致するか否か（一致 / 不一致）という情報が信号 3 0 2 として得られ、算術復号化演算部 3 0 1 から出力され、図 3 の復号化ブロックにて使用される。この信号 3 0 2 は、復号中の着目画素値が予測値に一致する場合 1、一致しない場合 0 という値をとる。

【 0 0 5 3 】

復号化ブロックでは、符号化ブロックと同様に、予測状態メモリ 1 0 5 は周辺画素データのコンテキスト情報をアドレスとして、復号中の着目画素に対する予測値 1 0 6 を出力する。

【 0 0 5 4 】

該予測値 1 0 6 は、算術復号化演算部 3 0 1 から出力される一致 / 不一致信号 3 0 2 との

50

間で、排他的NORゲート305にて論理演算される。この論理演算の結果は復号化した着目画素の値であり、復号画素値が信号304として得られる。

【0055】

その他のブロックの動作は、図1の符号化ブロックと同じなので、説明を省略する。

【0056】

次に、本発明人が提案している特開平11-103257に記載の、確率推定値から更新後の確率推定値を得る手段を有する算術復号化ブロックの構成を図5に示す。図5において、501は、現在の確率推定値から次(更新後)の確率推定値を求める予測確率更新部、503は、予測値と確率推定値を格納する確率推定値メモリであり、その他の要素は、図3で同一番号を付した要素と同じ機能を有する。

10

【0057】

図5の構成図を図3の構成図と比較すると、確率推定値メモリ503と算術復号化演算部301との間に、確率推定部109が無いということが分かる。

【0058】

算術復号化演算部301の入出力信号は図3に示す構成の場合と全く同じであるため、確率推定部109が無くなった影響は、他の構成要素である予測状態メモリ105や予測状態更新部119に現われる。

【0059】

算術復号化演算部301に必要な確率推定値111は、新たに設けた予測確率推定値メモリ503から読み出す。そのため、確率推定部109が不要になる。

20

【0060】

確率推定値メモリ503に確率推定値を格納するには、初期化時に、初期状態値に対応した確率推定値を確率推定値メモリ503に書き込み、該メモリ更新時に、更新後の確率推定値を書き込む。

【0061】

更新後の確率推定値は、予測確率更新部501で求める。予測確率更新部501は予測状態更新部119の替わりに用いるもので、現在の確率推定値、予測値等を入力して、更新後の確率推定値、予測値を求めるものである。現在の確率推定値から更新後の確率推定値への変換はROMテーブルあるいはデコーダ等を用いて行なう。

【0062】

よって、上記構成を有する復号化ブロックにおいては、状態値107が物理的には存在しなくなる。しかし、現在の確率推定値から更新後の確率推定値への変換テーブル(ROMテーブルあるいはデコーダの作成に必要となる)は、状態値の推移を確率推定値の推移に置き換えて作成するため、変換テーブルに内包されることになる。

30

【0063】

状態値の番号には一意性がある。すなわち、異なる状態には、必ず異なる状態値番号が割り当てられる。しかし、確率推定値には一意性があるとは言えない。すなわち、異なる状態に対して同一の確率推定値を割り当てることが原理的に可能である。しかし、JBIG方法においては、異なる状態に対して異なる確率推定値が割り当てられているので、JBIG方法に限れば確率推定値に一意性があるといえる。

40

【0064】

確率推定値に一意性があれば、予測確率更新部501で行う処理に何も問題はないが、一意性がない場合には問題が生じる。すなわち、入力確率推定値に対して更新確率推定値が何種類も存在してしまうという問題である。これを避けるには、同じ確率推定値を区別するための識別信号が必要になる。同じ確率推定値をとるものが2n個ある場合は、識別のためにnビットの識別信号が必要となる。この識別信号は予測確率更新部501におけるROMテーブルあるいはデコーダ等に入力され、何種類もある更新確率推定値のうちの一つを特定する。さらに出力側の更新確率推定値が一意でない場合には、この確率推定値を識別するnビットの信号が必要となり、識別信号は確率推定値メモリ503にも格納される。

50

【0065】

図1及び図3に示す構成を有する符号化/復号化ブロックにおける、1画素を符号化/復号化するのに要する主な処理は、以下の4ステップになる。

【0066】

- (1) 予測値と状態値をメモリから読み出す。
- (2) 状態値を確率推定値に変換する。
- (3) 確率推定値を用いて算術符号化/復号化演算を行なう。
- (4) 予測値と状態値を更新し、メモリに書き込む(正規化処理時のみ必要)。上記(4)における予測値と状態値の更新処理は、(3)における算術符号化演算と並列に行なうことができるので、タイミング的に(4)で行なう処理はメモリへの書き込み処理だけに

10

【0067】

これに対して、図5に示す構成を有する復号化ブロックでは、以下の3ステップになる。

【0068】

- (1) 予測値と確率推定値をメモリから読み出す。
 - (2) 確率推定値を用いて算術復号化演算を行なう。
 - (3) 予測値と確率推定値を更新し、メモリに書き込む(正規化処理時のみ必要)。
- 従って、図1及び図3に示す構成に比べて、状態値を確率推定値に変換する処理に相当する時間だけ、速く符号化することができる。

【0069】

また、本発明人はさらに特開平11-103257により、図6に示すような構成を有する算術復号化ブロックも提案している。

20

【0070】

一般に、予測値を求める基になるコンテキスト情報としては、着目画素と相関の強い真上及び左隣の画素を用いるため、復号化処理をラスタ走査順序で行なう場合、単純な逐次処理では、着目画素の復号化が終了しなければ、右隣の画素を復号化処理するために必要な予測値等を読み出すことができない。

【0071】

そこで、予測値等を格納したメモリを複数のメモリ群に分割して、該メモリ群から複数の予測値等を並列に読み出す。この複数の予測値は、復号化処理中の着目画素値が0と1の

30

【0072】

図6は、上記メモリ構成を有し、メモリに確率推定値を格納する算術復号化ブロックを示す。

【0073】

同図において、601は第1の確率推定値メモリ、602は第2の確率推定値メモリ、603は上記2つのメモリから読み出した2組の予測値・確率推定値の一方を選択するセレクタ、605はセレクタ603の出力と予測確率更新部501の出力のいずれか一方を選択するセレクタ、607はセレクタ605の出力をラッチするDタイプフリップフロップ(D-F/F)、610はコンテキスト情報、609はコンテキスト情報610を1サイクル遅延させる遅延回路である。その他の構成要素は、図5に示す復号化ブロックにおいて同一番号で示すものと同じであるため、説明を省略する。

40

【0074】

図6の構成では、予測値と確率推定値を先行してメモリから読み出すため、メモリ読み出し用のコンテキスト情報と、メモリの更新時に使用するメモリ書き込み用のコンテキスト情報が1サイクルずれ、別の信号になる。そこで、2つのコンテキスト情報(アドレス)を同時に入力できる2ポートメモリを用いるものとする。

【0075】

信号610は、復号化処理中の着目画素値を除く他のコンテキスト情報であり、これまで

50

に出てきたコンテキスト情報 103 に較べ、1ビット情報が少ない。コンテキスト情報 610 は、メモリ読み出し用のアドレス信号として、第1及び第2の確率推定値メモリ 601, 602 に与えられる。一方、コンテキスト情報 610 は、遅延回路 609 にて1サイクル遅延されて、メモリ書き込み用のアドレス信号として、第1及び第2の確率推定値メモリ 601, 602 に与えられる。

【0076】

第1の確率推定値メモリ 601 には、復号化処理中の着目画素値が0の時に対応する予測値と確率推定値を記憶しており、第2の確率推定値メモリ 603 には、復号化処理中の着目画素値が1の時に対応する予測値と確率推定値を記憶している。すなわち、2つのメモリ 601, 602 で記憶する全情報は、図5に示す確率推定値メモリ 503 で記憶する情報と同じである。

10

【0077】

第1及び第2の確率推定値メモリ 601, 602 から、コンテキスト情報 610 をアドレスとして、予測値と確率推定値を並列に読み出す。

【0078】

該メモリからの読み出しと並行して、着目画素の復号化処理（算術復号化演算）を行ない、メモリ読み出しデータが確定する頃に復号画素値が確定する。この復号画素値に基づいて、セクタ 603 にて2つのメモリ出力の一方を選択する。

【0079】

セクタ 603 の出力は、図5における確率推定値メモリ 503 から読み出した内容と基本的に等しいが、更新データを書き込む前に次の確率推定値を先行して読み出すため、更新データがセクタ 603 の出力に反映されない場合がある。

20

【0080】

それは、予測値・確率推定値を読み出して選択する元となった全コンテキスト情報と、メモリ更新時に用いる全コンテキスト情報が完全に一致した場合である。これは、（読み出しアドレス = 書き込みアドレス）かつ（現サイクルの復号画素値 = 1サイクル前の復号画素値）かつ（メモリ更新時）という条件に言い替えられる。

【0081】

上記条件が成り立つことを不図示の検出器で検出する場合には、セクタ 605 によりメモリ出力を更新データに切り換える。セクタ 605 の出力は、D - F / F 607 でラッチして、算術復号化演算部 301 に与え、次の画素を復号化するための算術演算が行なわれる。

30

【0082】

該算術演算をしている間、復号化中の画素値を含まない、1ビット少ないコンテキスト情報 610 に基づき2つの確率推定値メモリ 601, 602 から新たな確率推定値を読み出す。

【0083】

上記構成では、上述したように復号化のための算術演算と確率推定値メモリ 601, 602 からの読み出しを並行して行なうことができるので、さらに高速な復号化処理が可能となる。

40

【0084】

以上は2値の画像データを符号化及び復号化する例について説明したが、次に、複数のビットプレーンで構成される多値画像を画素順序で符号化及び復号化する場合について説明する。画素順序とはプレーン順序と異なり、例えば2ビットプレーンでは図7(a)に示すように、 $\dots, \{i-1\}[2], \{i-1\}[1], \{i\}[2], \{i\}[1], \{i+1\}[2], \{i+1\}[1], \{i+2\}[2], \dots$ （ $\{ \}[1]$ は上位ビットプレーン、 $\{ \}[2]$ は下位ビットプレーンを表わす）のように、下位ビットと上位ビットを交互に符号化し、画素を逐次符号化するものである。

【0085】

この時に問題となるのは、各々のプレーンにおいて何を参照してコンテキスト情報とする

50

かということである。各々のプレーンを独立に符号化する場合には、上位ビットを符号化する時には上位ビットプレーンだけを、下位ビットを符号化する時には下位ビットプレーンだけを参照すればよい。

【0086】

ところが、下位ビットを符号化する時に上位ビット、それも同一画素の上位ビットを参照したい場合には、上位ビットを下位ビットよりも前に符号化する必要がある。これに対しては、前記符号化順序を図7(b)に示すように、 $\dots, \{i-1\}[1], \{i-1\}[2], \{i\}[1], \{i\}[2], \{i+1\}[1], \{i+1\}[2], \{i+2\}[1], \dots$ のように変更すればよい。該符号化順序の変更によって、復号化が可能となる。

10

【0087】

【発明が解決しようとする課題】

上記従来例による復号化ブロックを用いて、多値画像の複数ビットプレーンを算術復号化しようとする、ある着目ビットを復号化する時、復号化処理に必要なすべてのコンテキスト情報が確定するのは最悪、直前のビットを復号化してからである。そのため、直前のビットを復号化してからでないと学習メモリを読み出すことができないので、ハードウェアで復号化処理を行なう場合、復号化動作速度を高速化する大きな妨げになっていた。

【0088】

本発明は上記問題点を鑑みてなされたものであり、単純な構成を有するハードウェアで、算術符号化された多値画像の算術復号化速度を高速化することを目的とする。

20

【0089】

【課題を解決するための手段】

上記目的を達成するために、1画素をmビット(m>2)で表される多値画像データを、各画素を構成する1ビット毎に算術符号化する本発明の算術符号化/復号化方法は、符号化/復号化対象画素を構成する各符号化/復号化対象ビットを符号化/復号化する為に、2サイクル以上前に符号化/復号化されたビットを用いてコンテキスト情報を生成するコンテキスト生成ステップと、前記符号化/復号化対象ビットを、前記コンテキスト情報に基づいて算術符号化/復号化する算術符号化/復号化ステップとを備え、前記多値画像データを構成するビットプレーンの数をm、該ビットプレーン番号をMSBから順に1、2、 \dots 、mとし、各画素の番号を{ }、ビットプレーンを[]で表すとすると、前記算術符号化/復号化するビットの順序として、{i}[j]の次は、j$\leq m$の場合{ i-1}[j+1]を、j=mの場合{i+m}[1]を算術符号化/復号化する。

30

【0091】

好適な一様態によれば、符号化/復号化する画素の数がnである場合、{i-1} 0及び{i+1}>nでは、所定値を符号化/復号化する。

【0092】

また、好適な別の一様態によれば、符号化/復号化する画素の数がnである場合、{i-1} 0及び{i+1}>nでは、符号化/復号化をスキップする。

【0093】

また、上記目的を達成するために、1画素をmビット(m>2)で表される多値画像データを、各画素を構成する1ビット毎に算術符号化する本発明の算術符号化/復号化装置は、符号化/復号化対象画素を構成する各符号化/復号化対象ビットを符号化/復号化する為に、2サイクル以上前に符号化/復号化されたビットを用いてコンテキスト情報を生成するコンテキスト生成手段と、前記符号化/復号化対象ビットを、前記コンテキスト情報に基づいて算術符号化/復号化する算術符号化/復号化手段とを備え、前記多値画像データを構成するビットプレーンの数をm、該ビットプレーン番号をMSBから順に1、2、 \dots 、mとし、各画素の番号を{ }、ビットプレーンを[]で表すとすると、前記算術符号化/復号化するビットの順序として、{i}[j]の次は、j$\leq m$の場合{ i-1}[j+1]を、j=mの場合{i+m}[1]を算術符号化/復号化する。

40

【0095】

50

好適な一様態によれば、算術復号化装置は、予測値及び確率推定値を保持する、各ビットプレーン毎に設けられた確率推定値メモリと、前記生成されたコンテキスト情報に基づいて予測値及び確率推定値を更新する予測推定更新手段とを有し、前記生成されたコンテキスト情報をアドレスとして、前記確率推定値メモリから予測値及び確率推定値を読み出す。

【0096】

また、好適な別の様態によれば、算術復号化装置は、予測値及び状態値を保持する、各ビットプレーン毎に設けられた予測状態メモリと、前記生成されたコンテキスト情報に基づいて予測値及び状態値を更新する予測状態更新手段とを有し、前記生成されたコンテキスト情報をアドレスとして、前記予測状態メモリから予測値及び状態値を読み出す。

10

【0098】

また、好適な一様態によれば、復号化する画素の数が n である場合、 $\{i-1\} = 0$ 及び $\{i+1\} > n$ では、所定値を符号化/復号化する。

【0099】

また、好適な別の様態によれば、復号化する画素の数が n である場合、 $\{i-1\} = 0$ 及び $\{i+1\} > n$ では、復号化をスキップする。

【0100】

【発明の実施の形態】

以下、添付図面を参照して本発明の好適な実施の形態を詳細に説明する。

【0101】

20

なお上述の通り、算術符号化方法の種類により使用されている用語が異なるため、本発明の実施の形態においても標準方法である JBIG 方法に用いられている用語を使用するが、これにより JBIG 方法の算術符号に限定されるものではない。

【0102】

[第1の実施形態]

本第1の実施形態では、1画素が2ビットからなる多値画像の算術符号化方法と該算術符号化方法に対応する算術復号化装置を示す。

【0103】

図8は、本第1の実施形態における符号化/復号化の順序を表わす図である。同図における矢印の示す順序で算術符号化/復号化をする。すなわち、 $\dots, \{i\}[1], \{i-1\}[2], \{i+1\}[1], \{i\}[2], \{i+2\}[1], \{i+1\}[2], \{i+3\}[1], \dots$ の順序で符号化/復号化する。ここで、 $\{ \ }[1]$ は上位ビットプレーンを、 $\{ \ }[2]$ は下位ビットプレーンを表わす。

30

【0104】

1番目の画素の最上位ビット即ち $\{1\}[1]$ の次は $\{-1\}[2]$ を符号化/復号化することとなるが、これは存在しない画素の下位ビットにあたる。また、画素数を n としたとき最後から2番目の画素の下位ビット即ち $\{n-1\}[2]$ の次は $\{n+1\}[1]$ を符号化/復号化することとなり、これも存在しない画素である。これらのビットは本来符号化する必要はないが、ダミービットとして例えば0を符号化しておき、復号化時には復号化されたダミービットを捨てて元画像を得てもよい。この方法によれば画像の先端及び

40

終端において例外的な動作を行なわなくても済むので装置構成が簡単になるという利点がある。

【0105】

上位ビットプレーンを符号化する時にはそれ以前に符号化した上位ビットのみを参照する。すなわち、上位ビットプレーンを符号化する時のコンテキスト情報はそれ以前に符号化した上位ビットのみを用いて構成する。

【0106】

下位ビットプレーンを符号化する時にはそれ以前に符号化した下位ビットと、該下位ビット以前に符号化した上位ビットとを参照する。具体的には上記順序で符号化を行う場合、 $\{i\}[2]$ を符号化する時に参照できる下位ビットは $\{i-1\}[2]$ であり、また上

50

位ビットは、参照する下位ビット $\{i - 1\} [2]$ を符号化する以前に符号化した上位ビット、すなわち $\{i\} [1]$ 以前の上位ビットである。実際的には $\{i - 1\} [2]$ 以前の下位ビットの一部と、 $\{i\} [1]$ とを参照する。

【0107】

従来は、復号対象となるビット情報を復号化する1つ前のサイクルで必要なコンテキスト情報が確定していたが、本第1の実施形態では、2つ前のサイクルで必要なコンテキスト情報が確定する。復号化順序は符号化順序とまったく同じであるため、図8に示すように $\dots, \{i\} [1], \{i - 1\} [2], \{i + 1\} [1], \{i\} [2], \{i + 2\} [1], \{i + 1\} [2], \{i + 3\} [1], \dots$ の順序で復号化処理が行なわれる。

10

【0108】

まず、上記復号化順序により、上位ビットプレーンの符号化データを復号する場合について考える。上位ビットの復号化にはそれ以前に符号化した上位ビットのみを参照するため、 $\{i + 1\} [1]$ を復号化するのに必要なコンテキスト情報は、 $\{i\} [1]$ 及びそれ以前の上位ビット情報である。この場合、 $\{i + 1\} [1]$ を復号化するのに必要な一番最後の情報である $\{i\} [1]$ の復号化は、 $\{i + 1\} \{i\}$ を処理するタイミングに対して2サイクル前に終了している。

【0109】

次に、下位ビットプレーンの符号化データを復号する場合について考える。 $\{i\} [2]$ を符号化する時に参照する上位ビット $\{i\} [1]$ の復号化は、 $\{i\} [2]$ に対し3サイクル前に終了している。また、下位ビットプレーン上の参照ビットの復号化は2サイクル前に終了している。従って、図8に示す復号化順序では、着目ビットの復号化処理に必要なコンテキスト情報は、いずれのプレーンにおいても2サイクル前には確定していることが解る。

20

【0110】

上記のように参照ビット情報を用いてコンテキストを生成し、符号化したデータをハードウェアで復号する方法においては、信号処理のタイミングが大変重要である。この場合は特にコンテキスト情報が確定するタイミングが重要となる。上記復号化順序で復号化処理可能な復号化ブロックの構成について、図9を参照して説明する。

【0111】

同図において、801は上位ビットプレーンを復号化する時のコンテキスト情報、803は上記コンテキスト情報801をラッチするD-F/F、805は下位ビットプレーンを復号化する時のコンテキスト情報、807はコンテキスト情報805をラッチするD-F/F、809は上位ビットプレーンを復号化する時に用いる確率推定値メモリ、811は下位ビットプレーンを復号化する時に用いる確率推定値メモリ、813は上記2つの確率推定値メモリの出力の一方を選択するセレクタ、815はセレクタ813の出力と更新した確率推定値の一方を選択するセレクタ、817はセレクタ815から出力する確率推定値をラッチするD-F/F、819は予測・確率更新部から出力する更新された予測値と確率推定値をラッチするD-F/Fである。なお、その他の構成要素は図6において同一番号で示す要素と同様の機能を有する。

30

40

【0112】

上記構成を有する復号化ブロックにおける復号化処理のタイミングを、画素ビット $\{i + 1\} [1]$ から画素ビット $\{i + 3\} [1]$ を処理する場合について、図10を参照して説明する。

【0113】

まず最初のサイクルで、 $\{i + 1\} [1]$ に対応する予測値及び確率推定値を確率推定値メモリ809から読み出す。セレクタ813は1サイクル毎に確率推定値メモリ809及び811を交互に選択するように制御されており、ここでは確率推定値メモリ809からの出力を選択する。また、セレクタ815は、図6のセレクタ615と同様に、(読み出しアドレス = 書き込みアドレス)かつ(現サイクルの復号画素値 = 1サイクル前の復号画

50

素値)かつ(メモリ更新時)の条件が成り立つ場合にのみD-F/F819の出力を選択する。従って、通常はスイッチ813の出力を選択している。読み出された予測値及び確率推定値はD-F/F817に一旦ラッチされる。

【0114】

次のサイクル2でD-F/F817にラッチされた予測値及び確率推定値は算術復号化演算部301に入力し、 $\{i+1\}[1]$ の符号化データが算術復号化される。この算術演算の結果、復号した画素値がコンテキスト情報の一部として得られるが、サイクルの後半に確定するので、このコンテキスト情報を安定した信号として使用するために、一度D-F/F803にラッチする。これによりD-F/F803から出力されるコンテキスト情報は1サイクル遅れた信号となるが、サイクルの先頭から安定した信号として使用することができる。

10

【0115】

また、該確率推定値も同じくサイクルの後半に確定するため、それを安定した信号として使用するためにD-F/F918でラッチする。

【0116】

また、サイクル2の間に、上記処理と平行して $\{i\}[2]$ に対応する予測値及び確率推定値を確率推定値メモリ811から読み出し、D-F/F817によりラッチする。

【0117】

次のサイクル3で、D-F/F819にラッチされた $\{i+1\}[1]$ に対する更新予測・確率推定値が正規化処理に対応して確率推定値メモリ809に書き込まれる。同時に、D-F/F817にラッチされた予測値及び確率推定値は算術復号化演算部301に入力し、 $\{i\}[2]$ の符号化データが算術復号化され、復号した画素値をコンテキスト情報の一部として、他の情報と一緒にD-F/F807によりラッチする。

20

【0118】

また、サイクル3の間に、上記処理と平行して $\{i+2\}[1]$ に対応する予測値及び確率推定値を確率推定値メモリ809から読み出し、D-F/F817によりラッチする。

【0119】

次のサイクル4では、D-F/F819にラッチされた $\{i\}[2]$ に対する更新予測・確率推定値が正規化処理に呼応して確率推定値メモリ811に書き込まれる。同時に、D-F/F817にラッチされた予測値及び確率推定値は算術復号化演算部301に入力する。なお、図10に示すように、 $\{i+2\}[1]$ の符号化データを復号化する際に用いられる予測値及び確率推定値はサイクル2で確定した $\{i+1\}[1]$ のコンテキスト情報に基づく値である。この予測値及び確率推定値を使用して $\{i+2\}[1]$ は算術復号化され、さらに、予測確率更新部501により得られる更新予測・確率推定値をD-F/F819によりラッチする。

30

【0120】

また、サイクル4の間に、上記処理と平行して $\{i+1\}[2]$ に対応する予測値及び確率推定値を確率推定値メモリ811から読み出し、D-F/F817によりラッチする。

【0121】

次のサイクル5では、D-F/F819にラッチされた $\{i+2\}[1]$ に対応する更新予測・確率推定値が正規化処理に呼応して確率推定値メモリ809に書き込まれる。同時に、D-F/F817にラッチされた予測値及び確率推定値は算術復号化演算部301に入力する。なお、図10に示すように、 $\{i+1\}[2]$ の符号化データを復号化する際に用いられる予測値及び確率推定値は、サイクル2で確定した $\{i+1\}[1]$ のコンテキスト情報及びサイクル3で確定した $\{i\}[2]$ のコンテキスト情報に基づく値である。この予測値及び確率推定値を使用して $\{i+1\}[2]$ は算術復号化され、さらに、予測確率更新部501により得られる更新予測・確率推定値をD-F/F819によりラッチする。

40

【0122】

また、サイクル5の間に、上記処理と平行して $\{i+3\}[1]$ に対応する予測値及び確

50

率推定値を確率推定値メモリ809から読み出し、D-F/F817によりラッチする。

【0123】

上記手順による処理を、すべての画素ビットについて繰り返し行う。

【0124】

このように、2サイクル先行していたコンテキスト情報としての復号画素値も、それを一度D-F/F803でラッチし、メモリ読み出し後にもう一度D-F/F817でラッチすることによって、2サイクルの遅れが生じ、最終的には、算術復号化演算部301における位相と一致する。

【0125】

上記の通り本発明の第1の実施形態によれば、多値画像を構成する複数ビットプレーンをプレーン間で数画素ずつずらして符号化する。すなわちあるビットプレーンを符号化する時に、コンテキスト情報として参照する別のビットプレーンを復号化対象画素ビットのビットプレーンより数画素先行して符号化し、復号化時は符号化時と同じ順序で復号化処理を行なうため、ハードウェアで復号化処理を行なう際に、復号化動作速度を大幅に高速化することが可能になる。

10

【0126】

[第2の実施形態]

本第2の実施形態では、1画素が4ビットからなる多値画像の算術符号化方法と該算術符号化方法に対応する算術復号化装置を示す。

【0127】

20

図11に本第2の実施形態における符号化復号化順序を表わす図を示す。同図における矢印の示す順序で算術符号化/復号化をする。すなわち、 \dots , $\{i\}[1]$, $\{i-1\}[2]$, $\{i-2\}[3]$, $\{i-3\}[4]$, $\{i+1\}[1]$, $\{i\}[2]$, $\{i-1\}[3]$, $\{i-2\}[4]$, $\{i+2\}[1]$, $\{i+1\}[2]$, $\{i\}[3]$, $\{i-1\}[4]$, $\{i+3\}[1]$, \dots の順序で符号化/復号化をする。ここで、 $\{ \ }[1]$ は最上位ビットプレーンを $\{ \ }[4]$ は最下位ビットプレーンを表わす。

【0128】

最上位ビットプレーンを符号化する時にはそれ以前に符号化した最上位ビットのみを参照する。すなわち、最上位ビットプレーンを符号化する時のコンテキスト情報はそれ以前に符号化した最上位ビットのみを用いて構成する。

30

【0129】

最上位以外のビットプレーンを符号化する時にはそれ以前に符号化した同一プレーンのビットと該ビット以前に符号化した上位のビットを参照する。上記符号化順序の場合、 $\{i\}[2]$ を符号化する時に参照できる同一プレーンのビットは $\{i-1\}[2]$ であり、また上位ビットは、参照する前記ビット $\{i-1\}[2]$ を符号化する以前に符号化した上位ビット、すなわち $\{i\}[1]$ 以前の上位ビットである。実際的には $\{i-1\}[2]$ 以前の同一プレーンのビットの一部と、 $\{i\}[1]$ とを参照する。

【0130】

ただし第1実施例と同様、画像の先端と終端付近では存在しない画素のビットを処理するタイミングが発生する。すなわち画像の先端では、 $\{1\}[1]$, $\{0\}[2]$, $\{-1\}[3]$, $\{-2\}[4]$, $\{2\}[1]$, $\{1\}[2]$, $\{0\}[3]$, $\{-1\}[4]$, $\{3\}[1]$, $\{2\}[2]$, $\{1\}[3]$, $\{0\}[4]$, $\{4\}[1]$, $\{3\}[2]$, $\{2\}[3]$, $\{1\}[4]$, という順番で処理を行なうとすると、2番目、3番目、4番目、7番目、8番目、12番目に処理を行うビットは実際には存在しない画素である。第1の実施形態と同様に0のダミービットを符号化しても良いが、本第2の実施形態ではビット情報がないところは符号化を行わず、復号化時に於いては後述する図12に示す有効画素タイミング発生回路1053とマスク回路1054によって確率推定値を0にしている。確率推定値が0であれば、算術復号化演算部301内のAレジスタ201及びCレジスタ202の値は変化しないので符号化処理を行なわないのと同じことにな

40

50

る。この構成はダミービットを符号化する方法に対して余計な符号を発生させないので符号長を短くできるという利点がある。

【0131】

上記第1の実施形態では、復号対象となるビット情報を復号化するのに必要なコンテキスト情報の2つ前のサイクルで確定していたが、本第2の実施形態では、必要なコンテキスト情報が4つ前のサイクルで確定する。復号化順序は符号化順序とまったく同じであるため、前記・・・、 $\{i\}[1]$ 、 $\{i-1\}[2]$ 、 $\{i-2\}[3]$ 、 $\{i-3\}[4]$ 、 $\{i+1\}[1]$ 、 $\{i\}[2]$ 、 $\{i-1\}[3]$ 、 $\{i-2\}[4]$ 、 $\{i+2\}[1]$ 、 $\{i+1\}[2]$ 、 $\{i\}[3]$ 、 $\{i-1\}[4]$ 、 $\{i+3\}[1]$ 、・・・の順序で復号化処理が行なわれる。

10

【0132】

まず、最上位ビットプレーンに着目して考える。最上位ビットの復号化にはそれ以前に符号化した最上位ビットのみを参照するため、 $\{i+1\}[1]$ を復号化するのに必要なコンテキスト情報は、 $\{i\}[1]$ 及びそれ以前の最上位ビット情報であり、一番間近の情報である $\{i\}[1]$ の復号化は $\{i+1\}[1]$ に対して4サイクル前に終了している。

【0133】

次に、他のビットプレーンについて考える。 $\{i\}[j+1]$ を符号化する時に参照した上位ビット $\{i\}[j]$ の復号化は、 $\{i\}[j+1]$ に対し5サイクル前に終了している。同一プレーン上の参照ビットは4サイクル前に終了している。従って、着目ビットの復号化処理に必要なコンテキスト情報は、いずれのプレーンにおいても4サイクル前には確定していることが解る。

20

【0134】

第2の実施形態が、ビットプレーンが2ビットから4ビットへ拡張されたことの他に第1の実施形態と異なる点は、メモリに保持する値としては状態値を用いること、バイパスの経路がないこと、有効画素のみを符号化するための手段が加えられていることである。

【0135】

上記のような参照ビット情報でコンテキストを生成し、符号化したデータをハードウェアで復号化する復号化ブロックの構成について、図12を参照して説明する。

【0136】

同図において、1010は最上位ビットプレーンを復号化する時のコンテキスト情報、1011はコンテキスト情報1010をラッチするD-F/F、1020は2番目に上位のビットプレーンを復号化する時のコンテキスト情報、1021はコンテキスト情報1020をラッチするD-F/F、1030は3番目に上位のビットプレーンを復号化する時のコンテキスト情報、1031はコンテキスト情報1030をラッチするD-F/F、1040は最下位ビットプレーンを復号化する時のコンテキスト情報、1041はコンテキスト情報1040をラッチするD-F/F、1012は最上位ビットプレーンを復号化する時に用いる予測状態メモリ、1022は2番目に上位のビットプレーンを復号化する時に用いる予測状態メモリ、1032は3番目に上位のビットプレーンを復号化する時に用いる予測状態メモリ、1042は最下位ビットプレーンを復号化する時に用いる予測状態メモリ、1050は上記4つの予測状態メモリ1012、1022、1032、1042の出力の一つを選択するセレクタ、1051は上記セレクタから出力する予測状態値をラッチするD-F/F、109は状態値を確率推定値LSZに変換する確率推定部、1053は、有効画素の確率推定値が変換されているとき1を出力する有効画素タイミング発生回路、1054は、無効画素に対応する確率推定値を強制的に0にマスクするマスク回路、1055はマスク回路から出力された確率推定値をラッチするD-F/F、1056は現在の確率推定値から更新時の状態値を予測する予測状態更新部、1057は予測状態の更新値をラッチするD-F/F、1058は復号したビット情報をラッチするD-F/Fである。なお、その他の構成要素は、図6において同一番号で示す要素と同様の機能を有する。

30

40

50

【0137】

状態値をメモリへ保持する構成が可能なのはコンテキストが4サイクル前に確定するので確率推定値を生成するために必要な余計なディレイを許容できるからである。例えば予測状態メモリ1012からある状態値が出力されたとすると、2サイクル後に算術復号化より復号ビット情報が排他的NORゲート305より出力されるので、D-F/F1058によりさらに1サイクル遅延を加えて4サイクル後に次の画素のコンテキストの一部としてD-F/F1011にラッチさせる。状態値は確率推定値よりビット幅が狭いので、メモリの容量を少なくし回路規模を小さくするためには本構成の方が好ましい。

【0138】

メモリへの更新データをバイパスする必要がないのは、各メモリからの読み出しが4サイクルおきであるためである。例えば予測状態メモリ1012からある状態値が出力され、さらに状態値の更新が行なわれるとすれば、その更新値がD-F/F1057にセットされるのは3サイクル後である。つまり3サイクル後にメモリ1012へ更新値の書き込み処理を行えば、次のサイクルで仮に4サイクル前と同じコンテキストが発生したとしても更新済みの状態値を読み出せる訳である。

【0139】

上記の処理タイミングを示したものが図13である。

【0140】

上記の通り本第2の実施形態によれば、多値画像を構成する複数ビットプレーンをプレーン間で数画素ずつずらして符号化する。すなわちあるビットプレーンを符号化する時に、コンテキスト情報として参照する別のビットプレーンを復号化対象画素ビットのビットプレーンより数画素先行して符号化し、復号化時は符号化時と同じ順序で復号化処理を行なうため、ハードウェアで復号化処理を行なう際に、復号化動作速度を大幅に高速化することが可能になる。

【0141】

なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェイス機器、リーダー、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0142】

【発明の効果】

以上説明したように本発明によれば、単純な構成を有するハードウェアで、算術符号化された多値画像の算術復号化速度を高速化することができる。

【図面の簡単な説明】

【図1】従来の算術符号化処理を行なう符号化ブロックの構成を示すブロック図である。

【図2】図1に示す算術符号化演算部の構成例を示すブロック図である。

【図3】従来の算術復号化処理を行なう復号化ブロックの構成を示すブロック図である。

【図4】図3に示す算術復号化演算部の構成例を示すブロック図である。

【図5】従来の算術復号化処理を行なう復号化ブロックの別の構成を示すブロック図である。

【図6】従来の算術復号化処理を行なう復号化ブロックの別の構成を示すブロック図である。

【図7】従来の符号化/復号化の順序を表わす図である。

【図8】本発明の第1の実施形態における符号化/復号化の順序を表わす図である。

【図9】本発明の第1の実施形態における算術復号化処理を行なう復号化ブロックの構成を示すブロック図である。

【図10】図9に示す復号化ブロックにより復号化を行う際の処理タイミングを示す図である。

【図11】本発明の第2の実施形態における符号化/復号化の順序を表わす図である。

【図12】本発明の第2の実施形態における算術復号化処理を行なう復号化ブロックの構成を示すブロック図である。

10

20

30

40

50

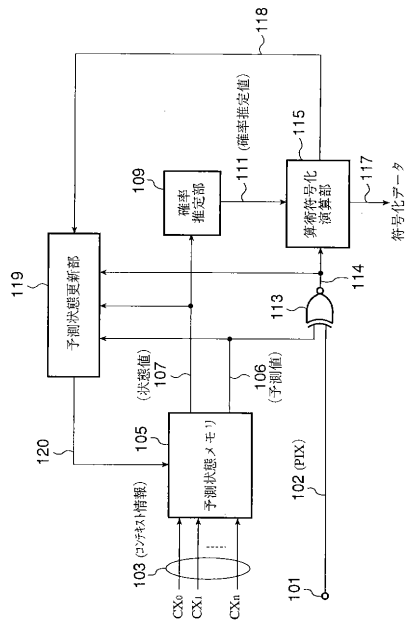
【図13】図12に示す復号化ブロックにより復号化を行う際の処理タイミングを示す図である。

【符号の説明】

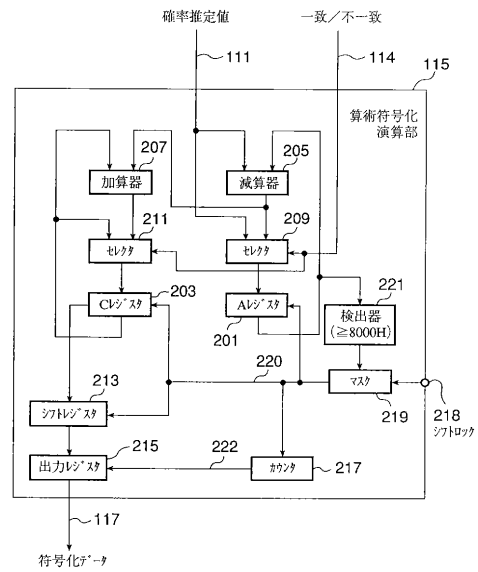
101	端子	
102	2値データPIX信号	
103	コンテキスト情報	
105	予測状態メモリ	
106	予測値	
107	状態値	
109	確率推定部	10
111	確率推定値LSZ	
113	排他的NORゲート	
114	一致/不一致信号	
115	算術符号化演算部	
117	符号化データ	
118	信号	
119	予測状態更新部	
120	制御信号	
201	Aレジスタ	
203	コードレジスタ	20
205	減算器	
207	加算器	
209	セレクタ	
211	セレクタ	
213	シフトレジスタ	
215	出力レジスタ	
217	カウンタ	
218	端子	
219	マスク回路	
220	シフトクロック	30
221	検出器	
301	算術復号化演算部	
302	信号	
304	復号画素値信号	
305	排他的NORゲート	
306	符号化データ	
401	減算器	
403	入力バッファレジスタ	
405	シフトレジスタ	
501	予測確率更新部	40
503	確率推定値メモリ	
601	第1の確率推定値メモリ	
602	第2の確率推定値メモリ	
603、605	セレクタ	
607	D - F / F	
609	遅延回路	
610	コンテキスト情報	
801、805	コンテキスト情報	
803、807、817、819	D - F / F	
809、811	確率推定値メモリ	50

- 813、815 セレクタ
- 1010、1020、1030、1040 コンテキスト情報
- 1011、1021、1031、1041 D - F / F
- 1012、1022、1032、1042 予測状態メモリ
- 1050 セレクタ
- 1051、1055、1057、1058 D - F / F
- 1053 有効画素タイミング発生回路
- 1054 マスク回路
- 1056 予測状態更新部

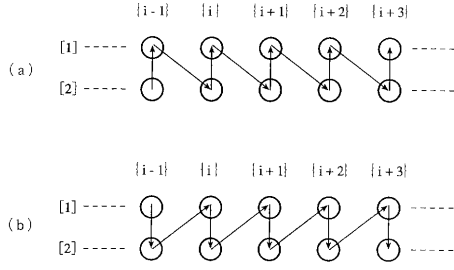
【図1】



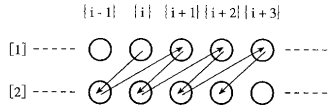
【図2】



【 図 7 】



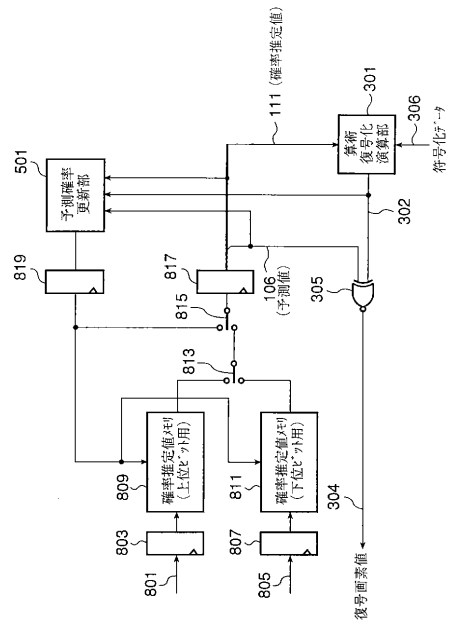
【 図 8 】



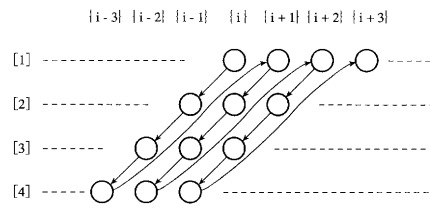
【 図 10 】

復号化対象画像	サイクル1	サイクル2	サイクル3	サイクル4	サイクル5
$i+1$ [1]	読み出し	復号化、 予測確率更新 (ラッチ)	予測確率書き込み		
i [2]		読み出し	復号化、 予測確率更新 (ラッチ)	予測確率書き込み	
$i+2$ [1]			$i+1$ [1]を 参照して 読み出し	復号化、 予測確率更新 (ラッチ)	予測確率書き込み
$i+1$ [2]				$i+1$ [1]及び i [2]を参照して 読み出し	復号化、 予測確率更新 (ラッチ)
$i+3$ [1]					$i+2$ [1]を 参照して 読み出し

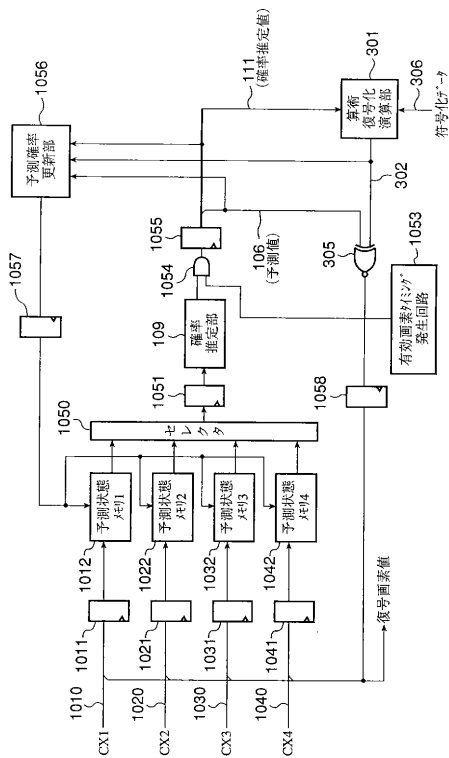
【 図 9 】



【 図 11 】



【図12】



【図13】

後方化対象要素	サイクル#1	サイクル#2	サイクル#3	サイクル#4	サイクル#5	サイクル#6	サイクル#7	サイクル#8
[1] [1]	読み出し	状態値を 更新し 読み出し	後方化 データ更新 読み出し	読み出し	読み出し	読み出し	読み出し	読み出し
[1] [2]			状態値を 更新し 読み出し	読み出し	読み出し	読み出し	読み出し	読み出し
[1] [3]			読み出し	読み出し	読み出し	読み出し	読み出し	読み出し
[1] [4]				読み出し	読み出し	読み出し	読み出し	読み出し
[1] [1] [1]					読み出し	読み出し	読み出し	読み出し
[1] [2]						読み出し	読み出し	読み出し
[1] [1] [3]							読み出し	読み出し

フロントページの続き

審査官 松永 稔

- (56)参考文献 特開平07-322070(JP,A)
特開平10-261074(JP,A)
特開2000-013609(JP,A)
特開平11-313212(JP,A)

(58)調査した分野(Int.Cl.⁷, DB名)

H04N 1/41
H03M 7/40
H04N 1/413
H04N 7/24