

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3871995号  
(P3871995)

(45) 発行日 平成19年1月24日(2007. 1. 24)

(24) 登録日 平成18年10月27日(2006. 10. 27)

(51) Int. Cl.		F I	
<b>HO4N</b>	<b>1/417</b>	<b>(2006.01)</b>	HO4N 1/417
<b>HO3M</b>	<b>7/42</b>	<b>(2006.01)</b>	HO3M 7/42

請求項の数 1 (全 20 頁)

(21) 出願番号	特願2002-311145 (P2002-311145)	(73) 特許権者	000005496
(22) 出願日	平成14年10月25日(2002.10.25)		富士ゼロックス株式会社
(62) 分割の表示	特願平9-165950の分割		東京都港区赤坂二丁目17番22号
原出願日	平成9年6月23日(1997.6.23)	(74) 代理人	100101948
(65) 公開番号	特開2003-198858 (P2003-198858A)		弁理士 柳澤 正夫
(43) 公開日	平成15年7月11日(2003.7.11)	(72) 発明者	土屋 徳明
審査請求日	平成14年10月25日(2002.10.25)		神奈川県海老名市本郷2274番地 富士
審判番号	不服2005-8853 (P2005-8853/J1)		ゼロックス株式会社内
審判請求日	平成17年5月12日(2005.5.12)		
		合議体	
		審判長	田口 英雄
		審判官	鈴木 明
		審判官	青柳 光代

最終頁に続く

(54) 【発明の名称】 符号化装置および復号化装置

(57) 【特許請求の範囲】

【請求項1】

画素ごとに順次振り分けられて入力されたデータを符号化する複数の符号化手段と、該複数の符号化手段で生成された符号データをそれぞれ格納する複数のメモリと、前記複数のメモリから前記符号データを読み出して単位符号データごとに所定の順序で1つの合成符号データに合成する合成手段を有し、該合成手段は、ある符号化手段において出力すべき符号データを格納する前記メモリが空で、かつ、他の符号化手段に対応する前記メモリが満杯の場合には識別子とともに他の符号化手段の出力する前記符号データを当該他の符号化手段に対応する前記メモリからすべて読み出して出力することを特徴とする符号化装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、予測符号化方式を用いた画像データの符号化装置およびその符号データの復号化装置に関するものである。

【0002】

【従来の技術】

2値画像の符号化技術の1つとして、周辺の画素から注目画素の状態を予測し、予測した結果を符号化する予測符号化方式がある。予測符号化においては、注目画素と相関関係があると思われる周囲画素をマルコフ情報源として予測に用いる。

## 【0003】

予測方式には、予測値を直接決定する方法と、間接的に決定する方法が知られている。予測値を直接決定する方法としては、例えば特開平5-316370号公報で示されるように、複数の予測器の中からの中率の高い予測器を適応的に切り替え、最終的に1つの予測器を選択し、原画像の予測に使用する方法がある。また、予測値を間接的に決定する方法としては、例えば、予測値等の情報をメモリに格納し、周辺の画素値からメモリの情報を読み出して予測に使用する方法がある。

## 【0004】

特公平6-95645号公報に開示されている予測方式は、後者のメモリを用いた間接的に予測値を決定する予測方式であり、周辺の複数画素の値をアドレスとして、テーブルルックアップ方式で出力される予測値を用い、予測誤差データを生成している。しかし、予測画素の位置を全て固定とはせず、1画素のみ位置を可変にし、予測的中率によって予測位置を変更させている方式をとっている。ITU-T T.82、T.85で定められたJBIG (Joint Bi-level Image coding expert Group) で用いられる予測方式は、上述の特公平6-95645号公報に示されている予測方式であり、符号化状態によって予測テーブルの情報を更新していく学習型の方式が採用されている。符号化には、注目画素を上述の予測方式で予測した結果である予測誤差データが使用される。

## 【0005】

JBIGにおいては、予測誤差データを符号化する符号化部に、算術符号化が採用された。算術符号化は、例えば特公平1-17295号公報に示されているように、数直線上の領域を入力されたシンボルの1, 0の発生確率に合わせて分割し、入力シンボルの値によって領域を選択する。このとき、予測と一致したシンボル(優勢シンボル)をMPS (More Probable Symbol)、一致しないシンボル(劣性シンボル)をLPS (Less Probable Symbol)と呼ぶ。このMPS、LPSの発生確率推定値は、予測的中率などによって変化させる。JBIGでは、シンボルの出現確率の推定部にLPS発生時の確率値(=領域幅LSZ)を持たせるとともに、その確率推定値をシンボルの出現過程に応じて更新する学習機能を持たせている。確率推定の学習機能は上述の予測値の学習機能と連動する。この分割選択処理を情報源の出力系列の分だけ行ない、最終段の分割で得られた領域内の1点の2進小数表示で出力系列を表わす。

## 【0006】

図10は、算術符号化の概念図である。斜線で示した領域が、それぞれシンボルの符号化終了後の有効領域である。初期状態では、有効領域は $[0, 1)$ 区間であり、LPS発生時の領域幅LSZ(0)が設定されている。 $1 - LSZ(0)$ がMPS発生時の領域幅である。ここではシンボルが「0」のときをMPS、「1」のときをLPSとする。図10において確率値は2進数で示している。

## 【0007】

図10では、最初に入力されたシンボルが「0」であるので、MPSの発生により有効領域は図10(A)においてハッチングを施した領域となる。そしてこの有効領域内をLPS発生時とMPS発生時の領域に分割する新たな領域幅LSZ(1)を設定する。次のシンボルが「1」のときは、LPSの発生により図10(B)においてハッチングを施した領域が有効領域となる。同様にして、3番目のシンボルが「0」のとき、図10(C)においてハッチングを施した領域が有効領域となり、さらに4番目のシンボルが「1」のとき、図10(D)においてハッチングを施した領域が有効領域となる。

## 【0008】

出力される符号はCで表わされる。入力されたシンボル列「0101」に対する符号は、図10(D)における有効領域内の1点とすればよい。ここでは、有効領域の基底、すなわち有効領域の最も小さい値C(3)を符号として表わし、出力する。

## 【0009】

算術符号化においては、符号化の過程で、有効領域の1点を符号とするため、領域を大き

10

20

30

40

50

くとれば、有効桁が減り符号量が減る。つまり、M P S 発生時にはなるべく L S Z の値を小さくし、L P S 発生時にはなるべく L S Z の値を大きくすることによって、有効領域を多く確保でき、符号量を減らすことができる。

#### 【 0 0 1 0 】

図 1 1 は、J B I G で採用されている予測符号化方式の概略図である。図中、1 は画像データ、2 はコンテキスト生成部、3 はコンテキスト、4 は予測部、5 はステート、6 は確率推定テーブル、7 は L S Z および S W、8 は予測値、9 は算術符号化部、1 0 は符号データ、1 1 は書き換えデータ、1 2 は書き換え信号である。

#### 【 0 0 1 1 】

画像データ 1 は、符号化のために算術符号化部 9 に入力されるとともに、コンテキスト生成部 2 に入力される。コンテキスト生成部 2 においては、注目画素の予測を行なうために、周辺画素を使用して予測のモデルテンプレートを作成する。図 1 2 は、J B I G で使用されるテンプレートの一例の説明図である。P を注目画素とすると、図 1 2 に示す参照画素  $X_0 \sim X_9$  を使用するテンプレートが生成される。この参照画素  $X_0 \sim X_9$  の値が  $2^{10}$  状態 (コンテキスト) の 1 0 重マルコフ情報源となり、そのまま予測部 4 にコンテキスト 3 として入力され、予測に使用される。

#### 【 0 0 1 2 】

予測部 4 には、参照画素  $X_0 \sim X_9$  の値に対応した予測値 8 と、確率推定テーブル 6 を参照するためのステート 5 とを対にした予測テーブルが記憶されている。参照画素  $X_0 \sim X_9$  の値であるコンテキスト 3 がそのまま予測テーブルにアドレスとして入力され、予測テーブルの 1 0 2 4 個のエントリのいずれかが選択される。コンテキスト 3 により選択された予測テーブルのエントリから予測値 8 およびステート 5 を取り出し、予測値 8 を算術符号化部 9 へ入力し、また、ステート 5 を確率推定テーブル 6 に入力する。図 1 3 は、予測部 4 に記憶されている予測テーブルの一例の説明図である。図 1 3 に示した予測テーブルは、アドレスに対応して、予測値とステートが対になって格納されている。なお、アドレスは 2 進数で示しているが、これはコンテキストコンテキスト 3 として入力される参照画素  $X_0 \sim X_9$  のそれぞれの画素値に対応するものである。例えば、参照画素  $X_0$  の値のみが「1」で他の参照画素は「0」であれば、アドレス「0 0 0 0 0 0 0 0 0 1」が参照され、予測値「1」とステート「2」が得られる。

#### 【 0 0 1 3 】

確率推定テーブル 6 は、ステート 5 に対応して、L P S 発生時の領域幅 L S Z と、L P S 発生時の次のステート値である N L P S および M P S 発生時の次のステート値である N M P S、さらに正規化発生時に予測値を書き換えるか否かを示すフラグである S W が格納されている。入力されたステート 5 に対応する L S Z および S W 7 を算術符号化部 9 へ出力するとともに、N M P S、N L P S を書き換えデータ 1 1 の一部として予測部 4 へ送る。図 1 4 は、確率推定テーブル 6 の一例の説明図である。図 1 4 では、ステート 5 の番号に対応して、L S Z、N M P S、N L P S、S W を対にして格納している。例えばステートが「2」のとき、L S Z として「0 x 1 1 1 4」、N L P S として「1 6」、N M P S として「3」、S W として「0」が得られる。ここで、「0 x」で始まる数値は 1 6 進数である。

#### 【 0 0 1 4 】

算術符号化部 9 では、予測部 4 から入力される予測値 8 と画像データ 1 とを比較し、一致していれば符号化するシンボルを「0」、すなわち M P S が発生したものとし、不一致であれば符号化するシンボルを「1」、すなわち L P S が発生したものとする。そして確率推定テーブル 6 から入力される L S Z を用い、図 1 0 で概略説明したように算術符号化を行なって符号データ 1 0 を出力する。L P S が発生した場合には、確率推定テーブル 6 から入力される S W が予測値の書き換えを可能としている場合 (例えば S W が「1」の場合) に限り、予測値を反転した新予測値を書き換えデータ 1 1 の一部として予測部 4 に送るとともに書き換え信号 1 2 を送り、予測部 4 に記憶されている予測テーブルの予測値を書き換える。

10

20

30

40

50

## 【 0 0 1 5 】

J B I Gにおいては、算術符号化部 9 において、有効領域の有効桁を増やさないよう、有効領域はつねに 0 . 5 以上 1 未満となるように調整されている。この処理を正規化と呼ぶ。また L S Z の値は必ず 0 . 5 未満の値となるように確率推定テーブルを構成しておく。これにより、L P S 発生時には有効領域は 0 . 5 未満となり、必ず正規化が発生する。さらに、M P S 発生時においても、新有効領域は、現有効領域から、L S Z を減算して求めるため 0 . 5 未満となる可能性があり、正規化が発生するケースがある。この正規化が発生した場合に、予測部 4 に対して書き換え信号 1 2 を送る。予測部 4 では、M P S 発生時には確率推定テーブル 6 から与えられる N M P S を、コンテキスト生成部 2 から与えられているコンテキスト 3 の状態として予測テーブルに書き込む。また、L P S 発生時には、確率推定テーブル 6 から与えられる N L P S をコンテキスト 3 の状態として予測テーブルに書き込む。このようにして予測テーブルが更新された後、次の画像データ 1 についての処理を開始する。

10

## 【 0 0 1 6 】

上述の予測部 4 と確率推定テーブル 6 を順次書き換えてゆく学習型の予測 / 確率推定方式について、具体例を用いながらさらに説明する。図 1 5 は、入力される画像データの一例の説明図である。図 1 5 では、図 1 2 に示すテンプレートの部分の画像データ 1 のみを示している。図 1 5 に示すように、コンテキスト  $X_9 \sim X_0$  が「0 0 0 0 0 0 0 0 1 1」のとき、注目画素 P が 1 であった場合を考える。予測部 4 の予測テーブルおよび確率推定テーブル 6 は、それぞれ図 1 3、図 1 4 に示す内容が設定されているものとする。

20

## 【 0 0 1 7 】

まず、コンテキスト「0 0 0 0 0 0 0 0 1 1」がコンテキスト 3 として予測部 4 に入力される。この時、予測部 4 の予測テーブルのアドレス「0 0 0 0 0 0 0 0 1 1」には、図 1 3 に示すように、予測値「0」と状態値「8 0」が格納されている。予測テーブルから読み出された状態値「8 0」は、確率推定テーブル 6 に入力される。次に確率推定テーブル 6 の状態番号「8 0」が参照される。確率推定テーブル 6 には図 1 4 に示す内容が設定されているので、その状態番号「8 0」に対応する L S Z 値「0 x 5 8 3 2」が読み出され、算術符号化部 9 へ出力される。

## 【 0 0 1 8 】

このとき、注目画素 P の値は「1」であるが、予測部 4 の予測テーブルから読み出された予測値は「0」であるため、L P S が発生した、ということになる。L P S が発生した場合、確率推定テーブル 6 から読み出された L S Z 値に対応する領域が基本的に次の有効領域となる。L S Z は、0 . 5 未満であるから、0 . 5 以上 1 未満になるように正規化が行なわれ、状態番号「8 0」の N L P S の値「8 0」が予測部 4 の予測テーブルのアドレス「0 0 0 0 0 0 0 0 1 1」の状態値として上書きされる。また、確率推定テーブル 6 の状態番号「8 0」の S W が「1」であるから、予測値は反転されて「1」となり、予測部 4 の予測テーブルのアドレス「0 0 0 0 0 0 0 0 1 1」の予測値として上書きされる。図 1 6 は、書き換え後の予測部テーブルの一例の説明図である。このようにして図 1 6 に示すように、予測部のアドレス「0 0 0 0 0 0 0 0 1 1」の予測値および状態値が書き換えられる。この書き換えられた後の予測部を用いて次の画像データ 1 についての予測を行なうことになる。次に同じコンテキストが入力された場合は、予測値が 1 となっている。

30

40

## 【 0 0 1 9 】

この例では L P S が発生したが、M P S が発生した場合には、正規化を行なったときには予測部 4 の予測テーブルの状態値を N M P S 「8 1」に書き換える。書き換え後、次に同じコンテキスト 3 が入力された場合、状態値が「8 1」であり、このときの L S Z は図 1 4 から「0 x 4 d 1 c」となる。この値は、状態値「8 0」のときの L S Z 「0 x 5 8 3 2」よりも小さく、つまり L P S が発生する確率が小さいことを意味する。これは、前回の同じコンテキストの時の予測が当たったため、次に同じコンテキストが入力された場合、M P S が発生時の領域幅を広げておき、再度予測が当たった場合に有効領

50

域を広くとれるようにしたものであり、最終的な有効領域を増大させ、符号量を減少させることができる。

#### 【0020】

以上のようにして、確率推定テーブルにしたがって予測部を順次書き換え、予測的中率を向上させ、MPS発生数を増やすとともに、MPS/LPS発生時の領域幅を制御して、符号化効率を向上させている。

#### 【0021】

また、復号時の処理も符号化時と同様、領域分割を行ないながら実施される。符号化時と異なるのは、符号化時は予測値と注目画素の比較を行ない、LPS/MPSの領域を演算しながら求めるのに対し、復号化時には、符号が、LPS/MPSのどちらの領域に入っているかを演算で求め、次に周辺の復号済みの参照画素から注目画素を逆予測しながら求める。

10

#### 【0022】

図17は、JBIGで採用されている予測復号化方式の概略図である。図中、図11と同様の部分には同じ符号を付して説明を省略する。21は算術復号化部、22は逆予測部、23は逆予測値、24は書き換えデータである。コンテキスト生成部2および逆予測部22は、符号化時のコンテキスト生成部2および予測部4とまったく同一に動作する。逆予測部22は符号化時の予測部4と同一である。当然、確率推定テーブル6も符号化時と同一のものが使用されなければならない。

#### 【0023】

まず符号化時と同様に、復号化された画像データ1の注目画素の周辺の参照画素の値によりコンテキストが生成され、逆予測部22、確率推定テーブル6から、LSZと逆予測値23が算術復号化部21に出力される。

20

#### 【0024】

次に、符号データ10は算術復号化部21において、有効領域の比較が行なわれた後、逆予測値23を用いて画像データ1を出力する。図18は、算術復号化部の基本動作の説明図である。まず、現有効領域から確率推定テーブルより入力されたLSZ幅を減算し、(有効領域 - LSZ)

を得る。この値を境にMPS領域とLPS領域が分割される。次に入力された符号Cが、先に求めた(有効領域 - LSZ)の値より大きいかが比較する。すなわち、 $C > \text{有効領域} - \text{LSZ}$

30

の判定を行なう。この比較判定によって、注目画素がLPSであるかMPSであるかが判定できるので、入力された逆予測値から、注目画素のレベルを求める。すなわち、MPSであれば注目画素の値は逆予測値であり、LPSであれば逆予測値を反転して注目画素の値とすればよい。以上の処理により、基本的に注目画素を復号できる。

#### 【0025】

ただし、次の画素の復号のために、ここからさらに、新有効領域の決定、符号 - 有効領域の減算等の処理が必要となる。例えば図18に示した例では、符号データCがLPS領域に存在するので、新たな有効領域としてLPS領域を設定し、新たなLSZを設定して上述の処理を繰り返し行なう。また、符号化時と同様にして、予測部4に記憶されている予測テーブルの予測値および状態値の書き換えも行なわれる。

40

#### 【0026】

以上のように、JBIGにおいては、確率推定テーブルとして書き換え可能なメモリを用いて更新してゆく学習型の予測方式を用い、またエントロピー符号化部に算術符号化方式を用いることによって、従来のランレングス符号化等に比べて高能率な2値画像の符号化および符号データの復号が達成される。

#### 【0027】

しかし、JBIG方式では、高能率な符号化が達成される反面、処理が低速になるという問題がある。JBIGで採用されている予測テーブルを使用する学習型の予測方式は、予測値、状態を書き換えて最適な状態に近づけるため、予測的中率が向上する利点もあ

50

る反面、メモリのアクセスに時間がかかり、また、予測テーブルの書き換えが発生した場合には書き換えが終了した後でなければ予測値を読み出すことができず、高速化が難しい。

【0028】

特願平8-301889号および特願平8-288731号などにおいては、符号化時あるいは復号時に予測テーブルあるいは逆予測テーブルから複数のデータをあらかじめ読み出して、現画像の符号化結果あるいは復号結果によってデータを選択して高速化する方法が提案されている。また、例えば特開平8-154059号公報に示されるように、算術符号化自体を高速化する方法も提案されている。しかしいずれにせよ、1画素ごとに予測テーブルの書き換えが発生する可能性があるため、隣接する複数の画素を同時に処理するなどの高速化の手段がとれないのが現状である。

10

【0029】

符号化の処理を高速化するために、符号器を複数用いて並列に処理する方法が一般に考えられる。すなわち、入力画像を複数の領域に分割して、それぞれに対応する符号器で符号化するものである。しかし、この方式では、スキャナ等から入力画像がページの先頭から1画素ずつ転送される場合には対応できない。例えば入力画像を2分割して2つの符号器で符号化する場合、一方の領域の画素が連続して転送され、その間はその一方の領域に対応する符号器のみが動作し、その後他方の領域の画素の転送が開始されるまで、他方の領域に対応する符号器は動作できない。そのため、複数の符号器を備えているにもかかわらず、並列に処理することができない。並列に処理するため、画像をメモリに一旦格納してから転送することが考えられるが、メモリ量が増えるという問題がある。また伸長時に出力装置などリアルタイム出力する場合も、メモリに一旦格納する必要がある。

20

【0030】

また別の方式として、例えば米国特許第4,982,292号明細書などに記載されているように、入力画像を1画素おきに符号化する方式なども提案されている。しかしこの方式では、JBIGにおいて高能率符号化を実現するコンテキストを1画素おきに生成することによって予測効率が落ちてしまい、符号化効率に影響するという問題がある。

【0031】

また、上述したいずれの高速化のための方式においても、並列的に処理された符号が別々に出力されるため、符号を保持するメモリ管理が複雑になるという問題がある。

30

【0032】

別々に生成された符号データを1つに合成する方法としては、例えば特開平6-38048号公報などに記載されているように、それぞれの符号データの先頭に識別コードを挿入する方法がある。この方法によれば、符号としては1つになるが、それぞれの符号データ長を長くするとそれだけ余分なメモリを必要とするし、切り替えを細かくすると識別コード量が増大してしまうという問題があった。

【0033】

【発明が解決しようとする課題】

本発明は、上述した事情に鑑みてなされたもので、高速に高効率の符号化あるいは復号化を行なうことができるとともに、符号データ量を抑えて1つの符号データとして管理可能な符号データを生成する符号化装置およびその符号データを復号する復号化装置を提供することを目的とするものである。

40

【0034】

【課題を解決するための手段】

上記課題を解決するために、第1の発明(請求項1に記載)の符号化装置は、複数の符号化手段で符号化した符号データを合成手段で合成する際に、各符号化手段から出力される符号データを単位符号データごとに順に出力する。符号化手段から符号データが出力されない場合には符号データの順次出力ができなくなるので、その場合に限って識別コードを付して他の符号化手段からの符号データの出力を行なう。これによって合成手段から出力される合成符号データには、通常の状態では識別コードが出力されず、符号化手段から

50

符号データが出力されないという特別な場合のみに識別コードが出力されるだけであるので、合成符号データ全体のデータ量を抑えることができる。

【0036】

【発明の実施の形態】

図1は、本発明の符号化装置の実施の一形態を示すブロック構成図である。図中、図11と同様の部分には同じ符号を付して説明を省略する。31は符号化部、32はFIFOメモリ、33はメモリ状態信号、34はリード信号、35は単位符号データ、36は合成部、37は合成符号データである。なお、図1では符号化部31を2ブロック設けた例を示しており、一方をAブロック、他方をBブロックとして、各ブロックに関係する構成には‘ - ’の後にAまたはBを記して区別している。

10

【0037】

各符号化部31-A, Bは、予測部4、確率推定テーブル6、算術符号化部9から構成されており、コンテキスト生成部2を各符号化部31-A, Bに共通した構成として設けている。画像データ1は、共通して設けられているコンテキスト生成部2に入力されるとともに、それぞれの符号化部31-A, Bの算術符号化部9に入力される。コンテキスト生成部2で生成されたコンテキスト3-A, Bは、それぞれの符号化部31-A, Bの予測部4に入力される。

【0038】

また、各符号化部31-A, Bに対応して、各符号化部31-A, Bから出力される符号データ10-A, Bを一時的に格納するFIFOメモリ32-A, Bを設けている。このFIFOメモリ32-A, Bは、合成部36からのリード信号34-A, Bに従って、先に格納した符号データから順に、基本的には所定データ量を単位として読み出し、単位符号データ35-A, Bとして合成部36へ渡す。FIFOメモリ32-A, Bは、それぞれ複数単位の符号データを格納可能であり、空状態か満杯状態を示すメモリ状態信号33-A, Bを出力する。

20

【0039】

合成部36は、各FIFOメモリ32-A, Bからのメモリ状態信号33-A, Bに基づき、基本的にはFIFOメモリ32-AとFIFOメモリ32-Bに対して交互にリード信号34-Aまたはリード信号34-Bを送出して符号データを読み出し、単位符号データ35-Aまたは単位符号データ35-Bを得て、合成符号データ37として出力する。FIFOメモリ32-AまたはFIFOメモリ32-Bの一方が空の状態而他方が満杯状態となった場合には、満杯状態となった側の識別符号を挿入して満杯となったFIFOメモリから連続して単位符号データを読み出し、合成符号データ37として出力する。また、符号データの終了時には、FIFOメモリ32-AまたはFIFOメモリ32-Bに残っている符号データの読み出しおよび出力を行なう。

30

【0040】

次に、本発明の符号化装置の実施の一形態における動作の一例について説明する。図2は、画像データの一例の説明図である。図2(A)には $(n+1) \times (n+1)$ 画素からなる原画像を示しており、図中の上から下方向に各ラインの左端から順に右端までの各画素のデータが入力されるものとする。すなわち、 $P_{00}, P_{01}, \dots, P_{0n}, P_{10}, P_{11}, \dots, P_{1n}, P_{20}, \dots, P_{mn}$ の順で各画素の画像データ1が入力される。図1に示した構成では、符号化部を2ブロック設けているが、各符号化部31-A, Bはそれぞれ交互に、入力される画像データ1の符号化を行なう。すなわち、符号化部31-Aは図2(B)に示すように、 $P_{00}, P_{02}, \dots, P_{10}, P_{12}, \dots$ の画素(2番目の添字が偶数の画素)についての符号化処理を行なう。また、符号化部31-Bは図2(C)に示すように、 $P_{01}, P_{03}, \dots, P_{11}, P_{13}, \dots$ の画素(2番目の添字が奇数の画素)についての符号化処理を行なうように制御される。

40

【0041】

コンテキスト生成部2は、入力された画像データ1から、例えば図12に示したようなテンプレートに従ってコンテキスト3を生成する。例えば $P_{22}$ の画像データ1が入力された

50

とき、図2(D)に示す $P_{22}$ を除く10画素によってコンテキストを生成する。次の $P_{23}$ の画像データ1が入力されたときには、図2(E)に示す $P_{23}$ を除く10画素によってコンテキストを生成する。これらのコンテキストは、いずれの符号化部によって使用されるかに関係なく、一律に生成される。このように符号化部に関係なくコンテキスト3を生成することによって、予測的中率を向上させ、効率よく符号化できるようにしている。

【0042】

符号化部31-Aでは、2番目の添字が偶数の画素が入力されると、コンテキスト生成部2から出力されるコンテキスト3-Aおよび画像データ1-Aを取り込む。例えば、 $P_{22}$ の画像データが入力されたとき、これを画像データ1-Aとして取り込むとともに、 $P_{22}$ に対応する図2(D)に示すテンプレートによって生成されたコンテキスト3-Aを取り込む。同様に、符号化部31-Bでは、2番目の添字が奇数の画素が入力されると、コンテキスト生成部2から出力されるコンテキスト3-Bおよび画像データ1-Bを取り込む。例えば、 $P_{23}$ の画像データが入力されたとき、これを画像データ1-Bとして取り込むとともに、 $P_{23}$ に対応する図2(E)に示すテンプレートによって生成されたコンテキスト3-Bを取り込む。

10

【0043】

各符号化部31-A, Bでは、上述の図11と同様の動作によって、コンテキスト3-A, Bに従って画像データ1-A, Bを適応的に予測符号化し、符号データ10-A, Bを出力する。

【0044】

図3は、本発明の符号化装置の実施の一形態における画像の入力と各符号化部の動作の一例を示すタイミングチャートである。入力画像クロックとともに1画素分の画像データ1が入力される。符号化部31-Aおよび符号化部31-Bには、入力画像クロックを2分周した、それぞれ位相の異なるクロックが入力される。符号化部31-A, Bにおいてはそれぞれのクロックの立ち上がりで画像データ1をラッチするため、結果的にラッチされた画像データA, Bは1画素おきにそれぞれの符号化部に入力されることになる。図3の例では、まず画素 $P_{20}$ が入力されている間に符号化部31-Aのクロックが立ち上がり、 $P_{20}$ が符号化部31-Aに取り込まれて符号化される。次に画素 $P_{21}$ が入力されている間に符号化部31-Bのクロックが立ち上がり、 $P_{21}$ が符号化部31-Bに取り込まれて符号化される。 $P_{21}$ が入力されている間は符号化部31-Aのクロックは立ち上がらないため、 $P_{21}$ は符号化部31-Aには取り込まれない。このようにして、符号化部31-Aと符号化部31-Bに交互に画像データ1が取り込まれて符号化される。

20

30

【0045】

図3に示したように、符号化部31-A, Bは、入力画像クロックの2周期分だけの間、符号化処理を行なうことができる。逆に、符号化部31-A, Bは並列的に動作するので、処理速度の倍の速度で画像データ1を入力して符号化することができ、全体の符号化速度は符号化部が1個の場合に比べて倍になる。このとき、コンテキスト生成部2を共用しているため、符号化部31-Aと符号化部31-Bはそれぞれ独立して符号化処理を行なっているにもかかわらず、的中率の高い予測を行なうことができ、圧縮効率を向上させている。

40

【0046】

各符号化部31-A, Bで符号化されて出力された符号データ10-A, Bは、それぞれ、FIFOメモリ32-A, Bに格納される。このときFIFOメモリ32-A, Bには、所定のデータ量を単位として複数単位の符号データが格納される。ここでは一例としてワード単位で符号データが格納されるものとする。

【0047】

図4は、本発明の符号化装置の実施の一形態における合成部の動作の一例を示すフローチャートである。このフローチャートでは、FIFOメモリ32-Aを'A'として示し、FIFOメモリ32-Bを'B'として示している。合成部36は、基本的にはS71~S74の動作を繰り返す。すなわち、S71においてメモリ状態信号33-Aを参照し、

50

F I F Oメモリ32 - Aが空か否かを判定し、空でなければS72においてF I F Oメモリ32 - Aから1ワード分の符号データ(単位符号データ35 - A)を読み出して合成符号データ37として出力する。次にS73においてメモリ状態信号33 - Bを参照し、F I F Oメモリ32 - Bが空か否かを判定し、空でなければS74においてF I F Oメモリ32 - Bから1ワード分の符号データ(単位符号データ35 - B)を読み出して合成符号データ37として出力する。これを繰り返すことによって、符号化部31 - Aと符号化部31 - Bで符号化した符号データが1ワードごとに交互に出力されることになり、符号化部31 - A, Bで生成した符号データが合成されたことになる。このような合成を行なうことによって、符号化部31 - A, Bでそれぞれ生成した符号データを別々に管理することなく、1つの符号データとして管理することができるようになる。また、それぞれの符号データを識別する識別コードをなるべく排除することができ、合成時のデータ量の増大を抑制している。

10

**【0048】**

算術符号化方式では、1つの画像データの入力に対して1つの符号が出力されるとは限らず、例えば非常に効率よく符号化できる場合、多数の画像データが入力されないと符号は出力されない。また、図10を用いて説明したように、有効領域の基底の小数部分を符号とするため、場合によっては有効領域の変化によって桁上がりが発生する場合がある。例えば有効領域の基底が2進数で0.0111のとき、有効領域の分割によって例えば0.10001等に変更される場合がある。このような場合には、この桁上がりが発生する可能性を考慮して、符号を出力することはできない。

20

**【0049】**

このように符号データが出力されない状態が一方の符号化部で発生し、他方の符号化部からの符号データの出力がF I F Oメモリに蓄積されてゆく場合がある。例えば符号化部31 - Aから符号データが出力されない状態が続き、符号化部31 - Bからの符号データがF I F Oメモリ32 - Bに蓄積されている場合には、S75で符号化が終了したか否かを判断した後、S76においてF I F Oメモリ32 - Bが満杯か否かを判定する。満杯でない場合には、そのままS71へ戻る。これは、なるべく1ワードごとに交互に符号データを出力するフォーマットを崩さないためであり、符号化部31 - Aから符号データが出力された時点で再び交互に1ワードずつ読み出し、出力する処理を開始する。

**【0050】**

さらに符号化部31 - Aから符号データが出力されず、F I F Oメモリ32 - Bが満杯になると、符号化部31 - Bは符号データを出力できないため符号化処理を停止しなければならなくなる。このような事態を回避するため、メモリ状態信号33 - Aが空を示し、メモリ状態信号33 - Bが満杯を示している場合には、S77において、直前に出力した符号データと同じブロックからの符号データを連続して出力する旨を示す識別コードを出力するとともに、F I F Oメモリ32 - Bからすべてのワードを読み出して出力する。そしてS71へ戻り、再びF I F Oメモリ32 - Aに符号データが格納されるのを待つ。

30

**【0051】**

符号化部31 - Bから符号データが出力されない状態が続き、符号化部31 - Aからの符号データがF I F Oメモリ32 - Aに蓄積されている場合も同様である。S72でF I F Oメモリ32 - Aから1ワード分の単位符号データ35 - Aを読み出して出力した後、S73でF I F Oメモリ32 - Bが空であると判定される。S78で符号化処理が終了したか否かを判定した後、S79でF I F Oメモリ32 - Aが満杯か否かを判定する。F I F Oメモリ32 - Aが満杯でなければS73へ戻り、F I F Oメモリ32 - Bに符号データが格納されるのを待つ。F I F Oメモリ32 - Bが空のままF I F Oメモリ32 - Aが満杯となると、S80において、識別コードを出力するとともに、F I F Oメモリ32 - Aからすべてのワードを読み出して出力する。そしてS73へ戻り、再びF I F Oメモリ32 - Bに符号データが格納されるのを待つ。

40

**【0052】**

このように、F I F Oメモリが満杯の場合には識別コードとともに符号データを出力する

50

ことによって、合成符号データのデータ量が多少増加するものの、符号化部を停止させることなく、高速な符号化処理を行なうことができる。

【0053】

符号化すべき画像データ1の入力が無くなり、符号化処理が終了した場合にも、いずれかのFIFOメモリに符号データが格納されたまま、他のFIFOメモリに符号データが格納されるのを待つことになる。これを回避するため、S75およびS78で符号化処理の終了を判定する。S75で符号化処理の終了を判定した場合には、S71でFIFOメモリ32-Aが空であると判定されているので、FIFOメモリ32-Bにデータが残っていることになる。S81において、識別コードを出力した後、FIFOメモリ32-B内の符号データをすべて読み出し、出力する。また、S78で符号化処理の終了を判定した場合は、S73でFIFOメモリ32-Bが空であると判定されているので、S82において、識別コードを出力後、FIFOメモリ32-A内の符号データをすべて読み出し、出力する。このようにして、すべての符号データを出力することができる。

10

【0054】

図5は、合成符号データの一例の説明図である。上述のようにして合成部36から出力された合成符号データ37の一例を図5に示している。ここで、'A'はFIFOメモリ32-Aから読み出した単位符号データ35-Aであり、'B'はFIFOメモリ32-Bから読み出した単位符号データ35-Bである。基本的にFIFOメモリ32-AとFIFOメモリ32-Bから交互に1ワードずつ単位符号データ35-A, Bを読み出して出力するので、FIFOメモリ32-Aから読み出した単位符号データ35-Aと、FIFOメモリ32-Bから読み出した単位符号データ35-Bが、1ワードずつ交互に並んでいる。すなわち、FIFOメモリ32-Aから読み出される単位符号データ35-Aが順に $A_0, A_1, \dots$ であり、FIFOメモリ32-Bから読み出される単位符号データ35-Bが順に $B_0, B_1, \dots$ であったとすると、まず、 $A_0, B_0, A_1, B_1, \dots$ とそれぞれのブロックの符号が交互に出力される。

20

【0055】

$A_n$ を出力した後、符号化部31-Aから符号データ10-Aが出力されない期間が続き、その間に符号化部31-Bから符号データ10-Bが出力されているとする。 $B_n$ を出力した後は、FIFOメモリ32-Bが満杯となるまで符号データの読み出しは行なわれず、そのままFIFOメモリ32-Bへ格納されるだけである。この状態がさらに続き、FIFOメモリ32-Bが満杯となると、識別コードを出力し、FIFOメモリ32-B内の符号データをすべて読み出して出力する。なお、ここではFIFOメモリ32の容量を16ワードとしている。図5に示すように、識別コードに続き、 $B_{n+1}$ から $B_{n+16}$ が出力されている。

30

【0056】

この例ではその後、符号化部31-Aから符号データの出力があり、 $A_{n+1}, B_{n+17}, \dots$ と再び交互に読み出しおよび出力がなされている。もちろん、その後も符号化部31-Aから符号データの出力がなく、再びFIFOメモリ32-Bが満杯となれば、識別コードとともにFIFOメモリ32-B内のすべての符号データを出力する。符号化部31-Bから符号データの出力がなく、FIFOメモリ32-Aに符号データが蓄積されてゆく場合も同様である。このようにして、符号化部31-Aと符号化部31-Bで別々に符号化した符号データを1つにまとめることができ、しかもデータ量の増加を抑えた合成符号データ37を出力することができる。

40

【0057】

上述の識別コードは、通常の符号と異なるものであれば何でもよい。例えば、JBIGの規格で定められるFF04(16進)などのコメントマーカを使用してもよい。

【0058】

図6は、本発明の復号化装置の実施の一形態を示すブロック構成図である。図中、図17と同様の部分には同じ符号を付して説明を省略する。41は復号化部、42はFIFOメモリ、43はメモリ状態信号、44はライト信号、45は単位符号データ、46は分割部

50

、47は合成符号データである。なお、図6では復号化部41を2ブロック設けた例を示しており、一方をAブロック、他方をBブロックとして、各ブロックに係る構成には‘-’の後にAまたはBを記して区別している。

【0059】

分割部46は、合成符号データ47を分割して、単位符号データ45-A, Bとするとともに、メモリ状態信号43-A, Bを参照しながらFIFOメモリ42-A, Bを制御して、単位符号データ45-A, BをそれぞれFIFOメモリ42-A, Bに書き込む。分割部46は、上述の符号化装置における合成部36と全く逆の動作を行なう。すなわち、基本的には合成符号データ47を所定のデータ量ごとに分割し、FIFOメモリ42-AまたはFIFOメモリ42-Bに交互に書き込む。また、識別コードを検出した場合は、

10

【0060】

FIFOメモリ42-A, Bは、復号化部41-A, Bにそれぞれ対応してその前段に設けられ、ライト信号44-A, Bに応じて所定のデータ量の単位符号データ45-A, Bを格納する。復号化部41-A, Bは、対応するFIFOメモリ42-A, Bから符号データ10-A, Bを読み出して復号する。

【0061】

復号化部41-A, Bは、図17に示した構成のうちコンテキスト生成部2を除く算術復号化部21、確率推定テーブル6、逆予測部22からなる。コンテキスト生成部2は、2つの復号化部41-A, Bに共通して設けられている。復号化部41-A, Bから交互に

20

【0062】

次に、本発明の復号化装置の実施の一形態における動作の一例について説明する。例えば図5に示したような合成符号データ47が分割部46に入力される。分割部46は、基本的には合成符号データ47を所定のデータ量の単位、例えば1ワード単位に分割し、分割した単位符号データ45-A, Bをライト信号44-A, BとともにFIFOメモリ42-A, Bに交互に書き込む。書き込みに際しては、メモリ状態信号43-A, Bを参照し、

30

【0063】

例えば図5に示す合成符号データ47では、最初の1ワードの単位符号データ $A_0$ をFIFOメモリ42-Aに書き込み、次の1ワードの単位符号データ $B_0$ をFIFOメモリ42-Bに書き込む。同様にして $A_1, \dots, A_n$ をFIFOメモリ42-Aに、 $B_1, \dots, B_n$ をFIFOメモリ42-Bにそれぞれ順次書き込んでゆく。メモリ状態信号43-A, Bを参照し、FIFOメモリ42-A, Bのいずれかが満杯の状態の場合には、その時点で書き込みを一時停止する。

【0064】

識別コードを検出すると、それに続く所定数の単位符号データは一方のFIFOメモリに書き込む。図5に示す合成符号データ47の場合には、識別コードの前にFIFOメモリ42-Bに書き込みを行なっているので、識別コードに続く16ワードの単位符号データをFIFOメモリ42-Bに順次書き込んでゆく。この場合もメモリ状態信号43-Bを参照し、FIFOメモリ42-Bが満杯の状態の場合には書き込みを一時停止する制御を行なう。

40

【0065】

各復号化部41-A, Bにおける動作は上述の図17と同様であり、コンテキスト生成部2で生成されたコンテキスト3-A, Bを用いて、FIFOメモリ42-A, Bから読み出した符号データ10-A, Bを復号し、画像データ1-A, Bとして出力する。このとき、復号化部41-A, Bは、コンテキスト生成部2から出力されるコンテキストを、復

50

号する画素位置に応じて交互に取り込み、復号する。例えば図2(A)に示す画像を復元する場合、図2(B)に示す画素を符号化した符号データを復号する復号化部41-Aは、2番目の添字が偶数となる画素の画像データを復号する際のコンテキスト3-Aを取り込む。同様に、図2(C)に示す画素を符号化した符号データを復号する復号化部41-Bは、2番目の添字が奇数となる画素の画像データを復号する際のコンテキスト3-Bを取り込む。

【0066】

図7は、本発明の復号化装置の実施の一形態における復号化部の動作と画像データの出力の関係の一例を示すタイミングチャートである。復号化部41-A, Bには、それぞれ位相が異なるクロックが与えられており、このクロックに従って復号を行なう。復号処理によって、ある時点で復号結果である画像データが確定する。このとき、復号化部41-Aと復号化部41-Bはそれぞれ独立に復号処理を行なっており、復号化部が1つの場合に比べて倍の速度で復号することが可能である。

10

【0067】

復号化部41-Aと復号化部41-Bとは位相の異なるクロックに従って動作しているので、図7に示すように復号結果が確定するタイミングが異なる。これを復号化部41-A, Bに与えられているクロックの1/2の周期の出力画像クロックによって交互に出力させる。これによって、図2(B)に示した画素を復号した復号化部41-Aの復号結果と、図2(C)に示した画素を復号した復号化部41-Bの復号結果とが交互に選択されて、図2(A)に示す画像のデータが順次出力されることになる。

20

【0068】

また、このようにして復号化部41-A, Bの復号結果を交互に選択した画像データはコンテキスト生成部2にも入力される。これによってコンテキスト生成部2では、符号化時と同様に、図2(D)や(E)に示すような連続した画素を参照したコンテキストを生成することができる。

【0069】

ところが、コンテキスト生成部2において、前画素の復号が終了しないと現画素のコンテキストが生成できないため、現画素の復号を行なうことができないという問題がある。すなわち、一方の復号化部で復号処理が遅れると、他方の復号処理を開始することができず、処理速度が低下してしまう。これを回避するために、例えば上述の特願平8-288731号に開示したように、復号時に前画素がとりうるすべての値に従って逆予測を行ない、複数の逆予測データを生成しておく。そして、前画素の値が確定した時点でいずれかの逆予測データを選択し、復号処理を高速化する方式が提案されており、本発明でもこの技術を用いることができる。

30

【0070】

図6に示した本発明の復号化装置においては、それぞれの復号化部41-A, Bにおいて1画素おきに復号される。そのため、例えば復号化部41-Aで現画素を復号する際に、復号化部41-Aにおいて直前に復号処理した画素、すなわち原画像上では2画素前の画素と、他方の復号化部41-Bにおいて復号している画素、すなわち原画像上の1画素前の画素をコンテキストに使用する。復号化部41-Aで直前に復号した画素については、値は確定しているものの、逆予測部22内の逆予測テーブルの書き換えが発生している場合、書き換え完了まで逆予測が行なえない。また、復号化部41-Bで復号する画素については、並行して処理を行なっているので値も確定していない場合がある。そのため、現画素の復号を行なう際には、原画像上の1画素前と2画素前の画素について、とりうるすべての値の組み合わせに対応した複数の逆予測値を求めておく。そして、復号化部41-Aで直前に復号した画素の値と、復号化部41-Bで現在復号している画素の値が確定した時点で、これらの画素の値から複数の逆予測値の中から正しいものを選択すればよい。

40

【0071】

このような動作を実現する構成を以下に示す。図8は、本発明の復号化装置の実施の一形態における復号化部の別の例を示すブロック構成図である。図中、51~54は逆予測テ

50

ーブル、55は書き込み制御部、56は書き込み信号、57はライトアドレス、58は選択部、59は他復号化部からの画像データ、60は前画像データである。ここでは、1つの復号化部の構成のみを示している。図8に示す構成が図6における復号化部41-A、Bに適用される。

【0072】

この構成では、例えば図12に示すテンプレートの $X_1$ 、 $X_0$ を除いた $X_9 \sim X_2$ の8個の画像データからなるコンテキストを用いる。逆予測テーブル51~54は、それぞれが例えば図13に示すような構成のテーブルであり、RAM等の記憶装置に格納されている。この逆予測テーブル51~54のアドレスは、図13に示したものと異なり、8ビットで構成されている。逆予測テーブル51は、 $X_1$ 、 $X_0$ の画素を0,0と仮定した時の逆予測値およびステートを格納している。同様に、逆予測テーブル52~54は、 $X_1$ 、 $X_0$ の画素をそれぞれ0,1、1,0、1,1と仮定したときの逆予測値およびステートをそれぞれ格納している。8ビットのコンテキストによって4つの逆予測テーブル51~54からそれぞれ逆予測値およびステートが読み出されて、選択部58に入力される。

10

【0073】

選択部58は、算術復号化部21から出力される画像データ1を前画像データ60として取り込むとともに、他復号化部からの画像データ59を取り込み、これらに基づいて、逆予測テーブル51~54から出力されている逆予測値およびステートのいずれかを選択し、ステートを確率推定テーブル6へ、逆予測値23を算術復号化部21へ出力する。例えば前画像データ60が0で、他復号化部からの画像データ59が0の場合には、逆予測テーブル51の出力を選択する。同様に、前画像データ60と他復号化部からの画像データ59が0,1の場合には逆予測テーブル52の出力を選択し、1,0の場合には逆予測テーブル53の出力を選択し、1,1の場合には逆予測テーブル54の出力を選択する。

20

【0074】

書き込み制御部55は、逆予測テーブル51~54で利用するコンテキスト $X_9 \sim X_2$ を書き込みアドレスとして逆予測テーブル51~54に対して出力するとともに、書き換え信号12の書き換え指示に応じて、算術復号化部21から出力される画像データ1および他復号化部からの画像データ59の値から書き込みを行なうべき逆予測テーブル51~54のいずれかを選択して、選択した逆予測テーブルに書き込み信号56を送る。各逆予測テーブル51~54には、確率推定テーブル6から出力されるNMP SまたはNLP Sおよび算術復号化部21から出力される新逆予測値が書き換えデータ24として与えられているので、書き込み制御部55から書き込み信号56が送られた逆予測テーブルでは、同じく書き込み制御部55から送られてくる書き込みアドレスに書き換えデータ24が書き込まれる。

30

【0075】

図9は、本発明の復号化装置の実施の一形態における復号化部の別の例の動作の一例を説明するためのタイミングチャートである。図9では、図2における $P_{24}$ の位置の画素の復号を復号化部41-Aで行なう例を示している。復号化部41-Aでは、 $P_{24}$ の位置の画素を復号する直前に、 $P_{22}$ の位置の画素を復号する。書き込み制御部55は、 $P_{22}$ の位置の画素を注目画素とするコンテキスト $X'_9 \sim X'_2$ を保持している。

40

【0076】

$P_{22}$ の位置の画素を復号中に、 $P_{24}$ の位置の画素を注目画素とするコンテキスト $X_9 \sim X_2$ が逆予測テーブル51~54にアドレスとして入力される。逆予測テーブル51~54から、それぞれ $X_1$ 、 $X_0$ が00,01,10,11の場合の4種類の逆予測値、ステートが読み出されて選択部58に入力される。

【0077】

一方、復号化部41-Bにおいては $P_{23}$ の位置の画素の復号が、復号化部41-Aにおける $P_{22}$ の位置の画素の復号に半周期遅れて開始される。やがて復号化部41-Aにおける $P_{22}$ の位置の画素が復号されて画像データが出力され、これが前画像データ60として選択部58に入力される。その後、復号化部41-Bにおける $P_{23}$ の位置の画素が復号され

50

て画像データが出力され、他復号化部からの画像データ59として選択部58に入力される。この時点で $X_1$ 、 $X_0$ の値が00、01、10、11のいずれであるかが確定されるので、選択部58は逆予測テーブル51～54のいずれかの出力(逆予測値、ステート)を選択して、 $P_{24}$ の位置の画素を復号するための逆予測値およびステートとして算術復号化部21および確率推定テーブル6に渡す。これによって次の周期で $P_{24}$ の位置の画素の復号を開始することができる。

【0078】

また、 $P_{22}$ の復号によって逆予測テーブルの書き換えが発生した場合には、書き込み制御部55は、保持している $P_{22}$ の位置の画素を注目画素とするコンテキスト $X'_9 \sim X'_2$ を書き込みアドレス57として予測テーブル51～54に出力する。また、算術復号化部21から出力される $P_{22}$ の位置の画像データ1と、復号化部41-Bから出力される $P_{23}$ の位置の画像データである他復号化部からの画像データ59から書き換えるべき逆予測テーブルを特定し、その逆予測テーブルに対して書き込み信号56を送る。逆予測テーブル51～54には、確率推定テーブル6および算術復号化部21から、 $P_{22}$ の位置の画像データに関する書き換えデータ24が入力されているので、書き込み制御部55から書き込み信号56が与えられた逆予測テーブルは、書き込みアドレス57のデータを書き換えデータ24に書き換える。次の $P_{26}$ の位置の画素の復号では、書き換えられた逆予測テーブルを用いて4種類の逆予測値およびステートが出力されることになる。

【0079】

もちろん、復号化部41-Aから出力される $P_{22}$ 、 $P_{24}$ の位置の画像データ1は、復号化部41-Bに送られて他復号化部からの画像データ59として用いられ、同様な処理で復号が実施される。

【0080】

このようにして、逆予測テーブルの書き換え前でも逆予測テーブルの読み出しが可能であるので、算術復号化部21の処理と逆予測テーブル51～54の読み出しを並列的に実行することができ、復号処理を高速化することができる。

【0081】

なお、逆予測テーブル51～54は、他復号化部からの画像データ59に対する1画素前の値のみによる2つの逆予測テーブルとして構成し、算術復号化部21から画像データ1が出力された後にコンテキスト $X_9 \sim X_1$ を用いるようにしてもよい。また、3ポートメモリを用いて、2つあるいは1つのメモリで構成してもよい。

【0082】

ここでは復号化装置について図8に示すような逆予測テーブルの先読みを行なって高速化する例を示したが、例えば上述の特願平8-301889号に記載されているように符号化装置について予測テーブルの先読みを行なうように構成すれば、符号化装置においても予測テーブルの書き込みによる遅延を発生させずに符号化処理を行なうことができ、符号化を高速に行なうことができる。

【0083】

以上、本発明による符号化部あるいは復号化部を複数配置して並列処理を行なう符号化装置および復号化装置について、一例を示した。本発明は、これらの例に限られるものではない。例えば、符号化部、復号化部を2ブロックではなく、もっと多数の構成にしてさらに並列化することも可能である。また、合成部36において合成する際の符号データのデータ量の単位も1ワード以外でよく、任意のデータ量を単位として、順次合成してゆくことができる。また、識別コード以後に続ける特定の符号化部からの符号データの単位数も任意に設定することができる。これらは復号化装置との間で設定可能であればよく、分割部46は符号化時と同じ規則で合成符号データを分割すればよい。

【0084】

また、FIFOメモリ32およびFIFOメモリ42の容量も16ワードに限るものではなく、任意の容量でよい。もちろん、FIFOメモリ32とFIFOメモリ42の容量が異なってもよく、合成符号データの形式に合わせて合成部36と分割部46がFIFO

10

20

30

40

50

メモリ 32、FIFOメモリ 42を制御すればよい。

【0085】

またさらに、本発明は予測符号化方式の符号データを合成符号データに合成するものに限られるものではなく、異なる符号化方式の符号データをも合成するように構成することも可能である。

【0086】

以上のように本発明では、JBIGなどの高速化が困難な符号化方式においても高速化することができるため、高速リアルタイム処理が必要となるプリンタ、複写機などのシステムにも、JBIG等の高能率の符号化方式を応用し、データを圧縮することが可能となる。

10

【0087】

【発明の効果】

以上の説明から明らかなように、第1の発明の符号化装置によれば、並列処理した符号データを効率よく1つの符号に合成することができるため、合成された符号データのデータ量の増大を抑えるとともに、符号データの管理を簡易にすることができる。

【0088】

また、第2の発明の復号化装置によれば、効率よく合成された符号データを入力として分割手段で複数の復号化手段に分配入力し、複数の復号化手段で並列処理が可能な復号化装置を得ることができるという効果がある。

【0089】

このほか、予測符号化を複数の符号化手段で並列処理を行なうことができるので、高速に符号化を行なうことができる。このとき、前段に大きな画像メモリを必要とせず、またコンテキストを単体の符号化手段を用いる場合と同じように生成できるため、符号化効率を損なうこともなく符号化することができる。また、この符号化装置と同様に、前段に大きな画像メモリを必要とせず、複数の復号化手段で並列して復号処理を行なうことができるので、高速にリアルタイムで復号することができる。

20

【図面の簡単な説明】

【図1】 本発明の符号化装置の実施の一形態を示すブロック構成図である。

【図2】 画像データの一例の説明図である。

【図3】 本発明の符号化装置の実施の一形態における画像の入力と各符号化部の動作の一例を示すタイミングチャートである。

30

【図4】 本発明の符号化装置の実施の一形態における合成部の動作の一例を示すフローチャートである。

【図5】 合成符号データの一例の説明図である。

【図6】 本発明の復号化装置の実施の一形態を示すブロック構成図である。

【図7】 本発明の復号化装置の実施の一形態における復号化部の動作と画像データの出力の関係の一例を示すタイミングチャートである。

【図8】 本発明の復号化装置の実施の一形態における復号化部の別の例を示すブロック構成図である。

【図9】 本発明の復号化装置の実施の一形態における復号化部の別の例の動作の一例を説明するためのタイミングチャートである。

40

【図10】 算術符号化の概念図である。

【図11】 JBIGで採用されている予測符号化方式の概略図である。

【図12】 JBIGで使用されるテンプレートの一例の説明図である。

【図13】 予測部4に記憶されている予測テーブルの一例の説明図である。

【図14】 確率推定テーブル6の一例の説明図である。

【図15】 入力される画像データの一具体例の説明図である。

【図16】 書き換え後の予測部テーブルの一具体例の説明図である。

【図17】 JBIGで採用されている予測復号化方式の概略図である。

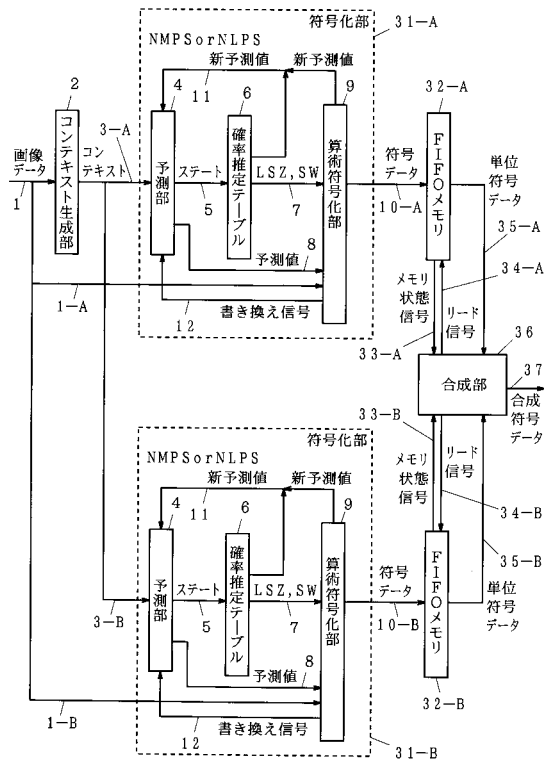
【図18】 算術復号化部の基本動作の説明図である。

50

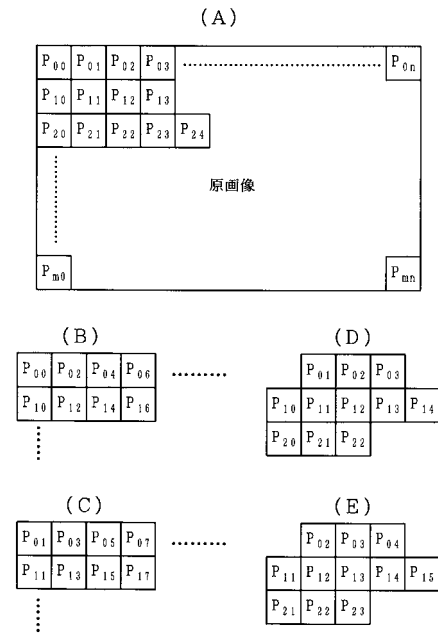
【符号の説明】

1 ... 画像データ、2 ... コンテキスト生成部、3 ... コンテキスト、4 ... 予測部、5 ... ステート、6 ... 確率推定テーブル、7 ... LSZおよびSW、8 ... 予測値、9 ... 算術符号化部、10 ... 符号データ、11 ... 書き換えデータ、12 ... 書き換え信号、21 ... 算術復号化部、22 ... 逆予測部、23 ... 逆予測値、24 ... 書き換えデータ、31 ... 符号化部、32 ... FIFOメモリ、33 ... メモリ状態信号、34 ... リード信号、35 ... 単位符号データ、36 ... 合成部、37 ... 合成符号データ、41 ... 復号化部、42 ... FIFOメモリ、43 ... メモリ状態信号、44 ... ライト信号、45 ... 単位符号データ、46 ... 分割部、47 ... 合成符号データ、51 ~ 54 ... 逆予測テーブル、55 ... 書き込み制御部、56 ... 書き込み信号、57 ... ライトアドレス、58 ... 選択部、59 ... 他復号化部からの画像データ、60 ... 前画像データ。

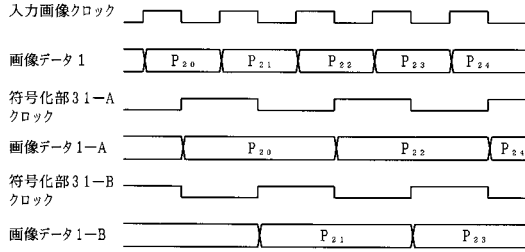
【図1】



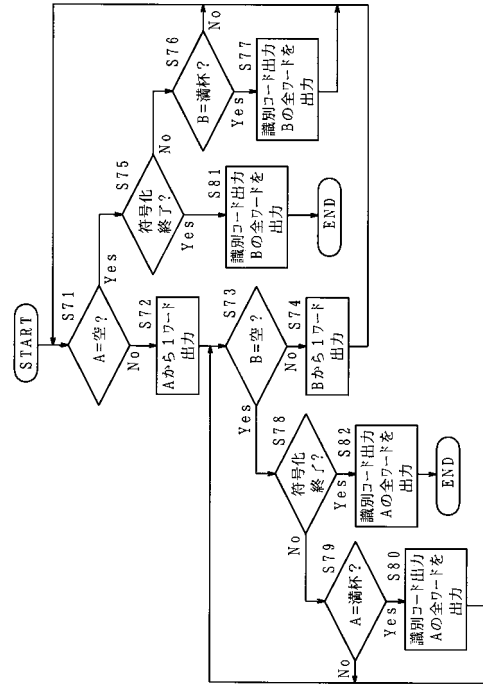
【図2】



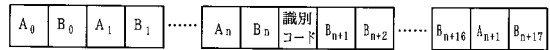
【図3】



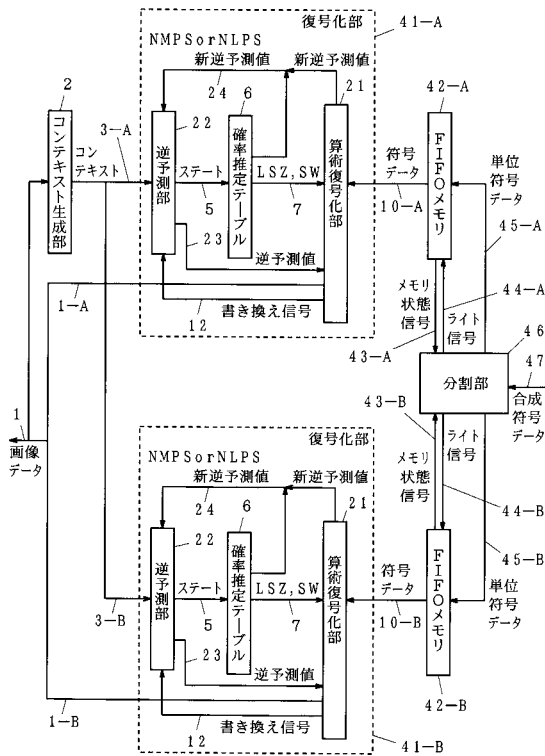
【図4】



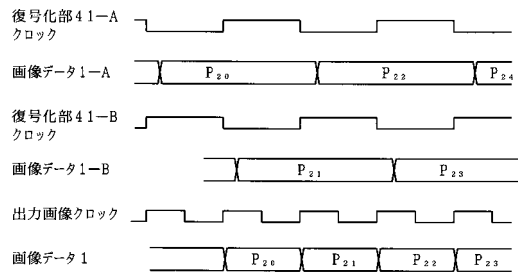
【図5】



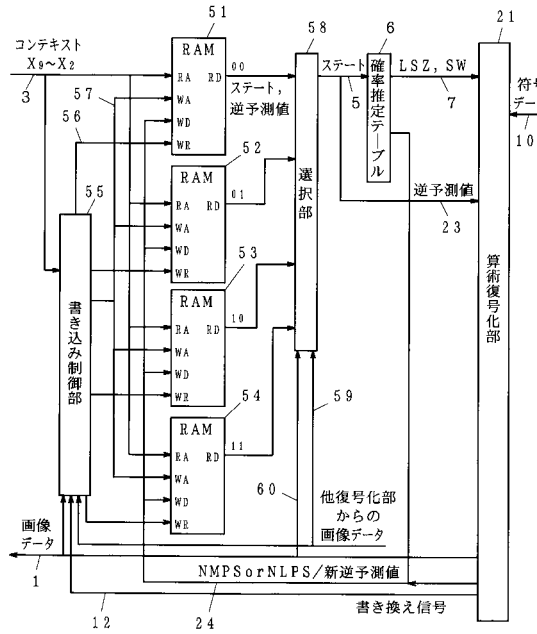
【図6】



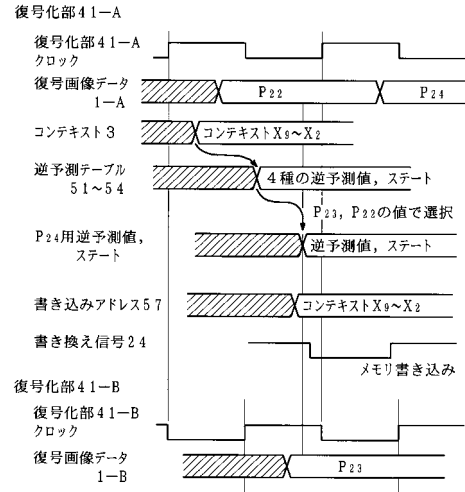
【図7】



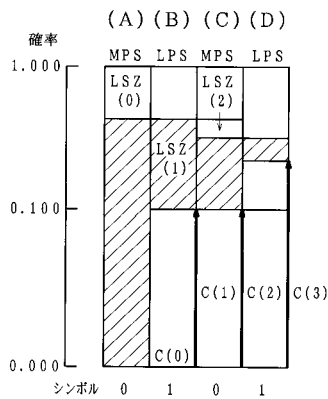
【図 8】



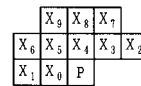
【図 9】



【図 10】



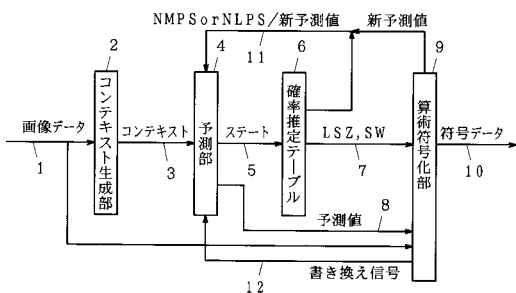
【図 12】



【図 13】

アドレス	予測値	ステート
0000000000	0	1
0000000001	1	2
0000000010	1	42
0000000011	0	80
⋮	⋮	⋮
1111111111	0	112

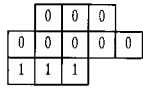
【図 11】



【図 14】

ステート	LSZ	NLPS	NMPS	SW
0	0x5a1d	1	1	1
1	0x2586	14	2	0
2	0x1114	16	3	0
⋮	⋮	⋮	⋮	⋮
80	0x5832	80	81	1
81	0x4d1c	88	82	0
⋮	⋮	⋮	⋮	⋮
112	0x59eb	112	111	1

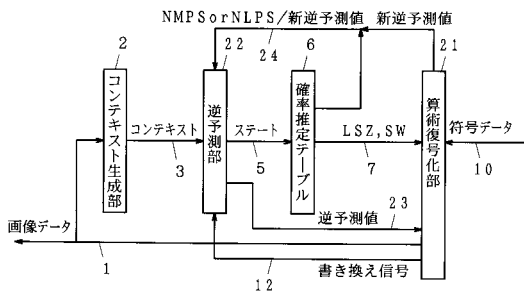
【 図 1 5 】



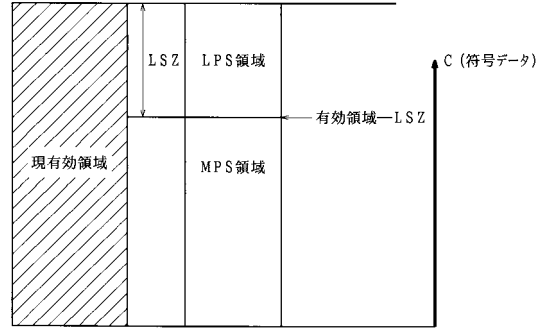
【 図 1 6 】

アドレス	予測値	ステート
0000000000	0	1
0000000001	1	2
0000000010	1	42
0000000011	1	80
:	:	:
1111111111	0	112

【 図 1 7 】



【 図 1 8 】



---

フロントページの続き

(56)参考文献 特開平9 - 288561 (JP, A)

特開平8 - 116534 (JP, A)

特開平8 - 107360 (JP, A)