

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第3840076号  
(P3840076)

(45) 発行日 平成18年11月1日(2006.11.1)

(24) 登録日 平成18年8月11日(2006.8.11)

(51) Int.C1.

F 1

|             |              |                  |        |       |   |
|-------------|--------------|------------------|--------|-------|---|
| <b>HO4N</b> | <b>1/413</b> | <b>(2006.01)</b> | HO 4 N | 1/413 | D |
| <b>HO4N</b> | <b>1/40</b>  | <b>(2006.01)</b> | HO 4 N | 1/40  | F |
| <b>HO4N</b> | <b>7/26</b>  | <b>(2006.01)</b> | HO 4 N | 7/13  | Z |

請求項の数 8 (全 45 頁)

|           |                              |
|-----------|------------------------------|
| (21) 出願番号 | 特願2001-259465 (P2001-259465) |
| (22) 出願日  | 平成13年8月29日 (2001.8.29)       |
| (65) 公開番号 | 特開2003-69835 (P2003-69835A)  |
| (43) 公開日  | 平成15年3月7日 (2003.3.7)         |
| 審査請求日     | 平成17年2月21日 (2005.2.21)       |

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

最終頁に続く

(54) 【発明の名称】画像処理装置及び方法及びコンピュータプログラム及び記憶媒体

## (57) 【特許請求の範囲】

## 【請求項1】

多値画像データの各画素に対する像域情報を入力し、圧縮符号化する画像処理方法であ  
って、

入力した前記像域情報を可逆符号化する第1符号化工程と、

該第1の符号化工程で可逆符号化された符号化データを伸長し、再度可逆符号化する第  
2符号化工程と、

前記第1符号化工程で得られる符号化データ量を監視し、該符号化データ量が所定基準  
を満たしたか否かを判断する監視工程とを備え、

該監視工程によって前記符号化データ量が所定基準を満たしたと判断した時、前記第1  
の符号化工程では、後続して入力される像域情報を所定の条件に従って変換してから可逆  
符号化し、前記第2の符号化工程では、第1符号化手段で既に符号化されている符号化デ  
ータが表す像域情報を所定の条件に従って変換してから再符号化することを特徴とする画  
像処理方法。

## 【請求項2】

更に、前記多値画像データを符号化する第3符号化工程を備え、

前記監視工程は、前記第3符号化工程で得られる符号化データ量と前記第1符号化工程  
で得られる符号化データ量の合計を監視し、前記合計のデータ量に基づいて前記所定基準  
を満たしたか否かを判断することを特徴とする請求項1に記載の画像処理方法。

## 【請求項3】

10

更に、前記多値画像データを符号化する第3符号化工程を備え、

前記監視工程は、前記第3符号化工程で得られる符号化データ量と前記第1符号化工程で得られる符号化データ量の合計を監視し、前記合計のデータ量及び前記第1符号化工程で得られる符号化データ量の夫々が目標符号量を越えたか否かを判断することにより、前記所定基準を満たしたか否かを判断することを特徴とする請求項1に記載の画像処理方法。

【請求項4】

多値画像データの各画素に対する像域情報を入力し、圧縮符号化する画像処理装置であって、

入力した前記像域情報を可逆符号化する第1符号化手段と、

該第1の符号化手段で可逆符号化された符号化データを伸長し、再度可逆符号化する第2符号化手段と、

前記第1符号化手段で得られる符号化データ量を監視し、該符号化データ量が所定基準を満たしたか否かを判断する監視手段とを備え、

該監視手段によって前記符号化データ量が所定基準を満たしたと判断した時、前記第1の符号化手段は、後続して入力される像域情報を所定の条件に従って変換してから可逆符号化し、前記第2の符号化手段は、第1符号化手段で既に符号化されている符号化データが表す像域情報を所定の条件に従って変換してから再符号化することを特徴とする画像処理装置。

【請求項5】

10

更に、前記多値画像データを符号化する第3符号化手段を備え、

前記監視手段は、前記第3符号化手段で得られる符号化データ量と前記第1符号化手段で得られる符号化データ量の合計を監視し、前記合計のデータ量に基づいて前記所定基準を満たしたか否かを判断することを特徴とする請求項4に記載の画像処理装置。

【請求項6】

20

更に、前記多値画像データを符号化する第3符号化手段を備え、

前記監視手段は、前記第3符号化手段で得られる符号化データ量と前記第1符号化手段で得られる符号化データ量の合計を監視し、前記合計のデータ量及び前記第1符号化手段で得られる符号化データ量の夫々が目標符号量を越えたか否かを判断することにより、前記所定基準を満たしたか否かを判断することを特徴とする請求項4に記載の画像処理装置。

30

【請求項7】

コンピュータが読み実行することで、多値画像データの各画素に対する像域情報を入力し、圧縮符号化する画像処理装置として機能するコンピュータプログラムであって、

入力した前記像域情報を可逆符号化する第1符号化手段と、

該第1の符号化手段で可逆符号化された符号化データを伸長し、再度可逆符号化する第2符号化手段と、

前記第1符号化手段で得られる符号化データ量を監視し、該符号化データ量が所定基準を満たしたか否かを判断する監視手段として機能し、

該監視手段によって前記符号化データ量が所定基準を満たしたと判断した時、前記第1の符号化手段は、後続して入力される像域情報を所定の条件に従って変換してから可逆符号化し、前記第2の符号化手段は、第1符号化手段で既に符号化されている符号化データが表す像域情報を所定の条件に従って変換してから再符号化することを特徴とするコンピュータプログラム。

40

【請求項8】

請求項7に記載のコンピュータプログラムを格納したことを特徴とする記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、画像データを該画像データに付随する像域情報を一定符号量以内に圧縮符号化

50

する機能を有する画像処理装置に関するものである。

【0002】

【従来の技術】

従来、静止画像の圧縮方式には、離散コサイン変換を利用したJPEG方式や、Wavelet変換を利用した方式が多く使われている。この種の符号化方式は、可変長符号化方式であるので、符号化対象の画像毎に符号量が変化する。

【0003】

国際標準化方式であるJPEG方式では、画像に対して1組の量子化マトリクスしか定義できないので、プリスキャン無しには、符号量調整が行なえず、限られたメモリに記憶するシステムで使用する場合においては、メモリオーバーを起こす危険性があった。

10

【0004】

これを防止するために、予定した符号量よりオーバーした場合は、圧縮率を変更して、原稿の再読み込みを行なう方法や、予めプリスキャンによる符号量見積もりを行ない、符号量を調整するために、量子化パラメータの再設定を行なう方法などとされていた。

【0005】

一方、画像情報には、元の画像データ以外に、該画像データに付随する像域情報というものがある。像域情報は主に、画像出力時の見栄えを良くするために、画像出力部での色処理や諧調数の調整に用いられる。有彩色と無彩色が混在する自然画像と、原稿中に多く見られる黒文字とでは、同じ黒色でも使用するインクの種類を変えることで、自然画像を自然画像らしく見せる一方で、鮮明な文字を出力することが出来る。

20

【0006】

このように、画素毎に、有彩色か無彩色、文字部かそうでないか、といった各々1ビットの属性フラグデータを持つことにより、画像出力時、特にプリントアウト時に出力画像の画質向上を図ることが出来る。像域情報には前記以外の他の情報もある。

【0007】

画像情報を圧縮するには、画像データの圧縮はもちろんのこと、上記像域情報も圧縮する必要がある。像域情報は2値データの集まりであり、これを圧縮するには基本的に可逆の符号化方式を用いる必要がある。従来、像域情報の圧縮にはP a c k b i t s やJ B I G符号化方式が用いられてきた。

【0008】

30

しかしながら、前記像域情報をこれらの符号化方式で圧縮するだけでは、符号量調整を行なえず、限られたメモリに記憶するシステムで使用する場合においては、メモリオーバーを起こす危険性があり、大きな問題であった。

【0009】

【発明が解決しようとする課題】

しかしながら、従来は、画像データの圧縮ばかりが検討され、像域情報の圧縮については、検討されることが少なかった。まして、該像域情報の圧縮後の符号量を目標値以内に収めること等、ほとんど考えられることが無く、ある符号化方式を用いて該像域情報を単純に符号化するだけであった。

【0010】

40

本発明は上記従来例に鑑みて成されたものであり、像域情報の入力を目標値以内の符号量に収めるための効果的な画像処理装置及びその制御方法及びコンピュータプログラム及び記憶媒体を提供しようとするものである。

【0011】

【課題を解決するための手段】

かかる課題を解決するため、例えば本発明の画像処理方法は以下の構成を備える。すなわち、

多値画像データの各画素に対する像域情報を入力し、圧縮符号化する画像処理方法であつて、

  入力した前記像域情報を可逆符号化する第1符号化工程と、

50

該第1の符号化工程で可逆符号化された符号化データを伸長し、再度可逆符号化する第2符号化工程と、

前記第1符号化工程で得られる符号化データ量を監視し、該符号化データ量が所定基準を満たしたか否かを判断する監視工程とを備え、

該監視工程によって前記符号化データ量が所定基準を満たしたと判断した時、前記第1の符号化工程では、後続して入力される像域情報を所定の条件に従って変換してから可逆符号化し、前記第2の符号化工程では、第1符号化手段で既に符号化されている符号化データが表す像域情報を所定の条件に従って変換してから再符号化することを特徴とする。

#### 【0012】

##### 【発明の実施の形態】

以下、添付図面に従って本発明に係る実施形態を説明するが、先ず、基本部分について説明する。

#### 【0013】

図1は、実施形態が適用する画像処理装置100の機能ブロック構成図である。以下、同図の各部を簡単に説明する。

#### 【0014】

画像処理装置100は、イメージスキャナから画像を入力する入力部101を備えている。なお、入力部101は、ページ記述言語レンダリングなどから画像データを入力しても良いし、記憶媒体に格納された画像ファイルを読み込むことで実現しても良く、場合によつてはネットワークより受信するようにしても良い。

#### 【0015】

符号化部102は、入力された画像データの符号化を行なう。なお、符号化方式は公知のJPEG符号化方式を用い、8×8画素単位に相当する画像データを直交変換し、後述する量子化ステップを用いた量子化、ハフマン符号化処理を行なうものである。

#### 【0016】

第1のメモリ制御部103と第2のメモリ制御部105は、上記符号化部102から夫々に出力されてくる上記符号化データ（同じ符号化データ）を第1のメモリ104と第2のメモリ106へ格納する様に制御する。ここで、第1のメモリ104は、最終的に確定した（目標値以内のデータ量に圧縮し終わった）符号化データを、図1の基本構成の外部に接続されるネットワーク機器、画像出力装置や大容量記憶装置等へ出力するために、該符号化データを保持するためのメモリである。また、第2のメモリ106は、前記符号化データを第1のメモリ上に形成するための圧縮符号化処理を補助する作業用のメモリである。

#### 【0017】

カウンタ107は、符号化部102によって圧縮符号化された画像データのデータ量をカウントし、該カウント値を保持すると共に、そのカウント結果を符号化シーケンスの制御を行なう符号化シーケンス制御部108に出力する。

#### 【0018】

符号化シーケンス制御部108では、カウンタ107のカウント値がある設定値に達したかどうかを検出し、その設定値に達した（目標値を越えた）ことを検出した時にメモリ104内の格納済みのデータを廃棄するよう第1のメモリ制御部103に制御信号を出力する。上記第1のメモリ制御部103は、この制御信号に基づいて、メモリアドレスカウンタをクリアするか、あるいは符号化データ管理テーブルをクリアすることにより、前記格納データを廃棄する。また、このとき、符号化シーケンス制御部108は、第1のカウンタ107をゼロクリアする（入力部101からの入力は継続している）と共に、符号化部102に対して今までより、高い圧縮率で符号化を行なうよう制御する。すなわち、本装置の符号化処理で発生する符号化データのデータ量が最終的に例えば1/2になるように制御する。なお、ここでは、1/2としたが任意に設定できることは言うまでもない。

#### 【0019】

そして、圧縮率変更後の符号化データも、これまでと同様、第1のメモリ制御部103と

10

20

30

40

50

第2のメモリ制御部105を経て、第1のメモリ104と第2のメモリ106に夫々格納される。

**【0020】**

さらに、符号化シーケンス制御部108は、第2のメモリ制御部105に対して、これまでに第2のメモリ106に格納した符号化データを読み出し、符号化データ変換手段である再符号化部109に該符号化データを出力するよう制御信号を出す。

**【0021】**

再符号化部109は、入力された符号化データを復号化し、データ量を減らすための再量子化等を行なった後に再び符号化処理を行ない、圧縮率が変更された符号化部102と同じ圧縮率のデータ量を第2のカウンタ110に出力する。 10

**【0022】**

この再符号化部109から出力される符号化データは、第1のメモリ制御部103と第2のメモリ制御部105を経由して、それぞれ、第1のメモリ104と第2のメモリ106に格納される。

**【0023】**

再符号化処理が終了したかどうかは、第2のメモリ制御部が検出する。すなわち、再符号化処理のために読み出すデータが無くなれば、再符号化処理の終了を符号化シーケンス制御部108に知らせる。実際には、第2のメモリ制御部105の読みだし処理だけでなく、再符号化部109の処理も終了した後に、符号化処理が完了したことになる。

**【0024】**

第2のカウンタ110で得られるカウント値は、再符号化処理が完了した後、第1のカウンタ107で保持されているカウンタ値に加算される。この加算結果は再符号化処理が完了した直後における、第1のメモリ104内のデータ量の合計を表す。即ち、1画面分の符号化部102と再符号化部109の符号化処理が終了した時点では、上記加算後の第1のカウンタ107で保持されているカウンタ値は、1画面分を本装置が符号化した場合に発生した総データ量を表す（詳細は後述）。 20

**【0025】**

符号化部102は、再符号化処理の終了／未終了に関わらず、符号化するべき入力部101からの画像データが残っている限りは符号化処理を継続して行なう。

**【0026】**

カウンタ107のカウント値がある設定値に達したかどうかは入力部101から入力される1ページ分の画像データの符号化処理（符号化、再符号化）が終わるまで繰り返され、上述した符号化と再符号化の処理は、ここで得られる検出結果に応じた制御の上で実行される。 30

**【0027】**

上記、図1の構成における処理のフローを表わすフローチャートを図8に示すが、説明を簡単にするため、簡略化した図3のフローチャートに従って先ず説明する。

**【0028】**

既に説明したように、本発明の画像処理装置100は、スキャナ等の入力部101から入力した1ページの画像データを所定のデータ量以下に圧縮符号化する装置である。該符号化処理を実現するために、前記入力部101以外に、符号化部102、再符号化部109、第1のメモリ104、第2のメモリ106等を有する。これらの機能ブロックを用い、図3に示すフローチャートに基づいて符号化処理を行なう。 40

**【0029】**

図3のフローチャートは、大別すると、下記の3つの処理フェーズに分かれる。

(1) 符号化フェーズ

(2) 符号化・再符号化フェーズ

(3) 転送フェーズ

上記それぞれの処理フェーズにおいて、どのように画像データ、符号化データ等が流れて処理され、メモリにどのように格納されるかを視覚的に解り易く示したのが図4乃至図7で

ある。

**【0030】**

図4は、図3のフローチャートにおけるステップS303とS305に対応する符号化フェーズの初期状態を表わす。また、図5はステップS307～S315に対応する符号化・再符号化フェーズの処理状態を、図6はステップS317に対応する転送フェーズの処理状態を、図7は転送フェーズ後の符号化フェーズの処理状態を表わす。以下、各フェーズについて説明する。

**【0031】**

<<符号化フェーズ>>

1ページ分の画像データの符号化処理は、符号化パラメータの初期設定（ステップS301）から始まる。ここでは符号化処理する画像サイズ（スキャナ等の入力部101から読み取る用紙サイズ）から一意的に定まる符号化データ量の上限値や符号化部102（ここでは公知のJPEG符号化方式を用いるものとする）に適用する量子化ステップ（Q1）といったパラメータを設定する。10

**【0032】**

そして、ステップS303にて、第1のカウンタ107は、実際の符号化処理（画像の8×8画素単位にJPEG圧縮）を行ない、出力される符号化データのデータ量を累積カウントする。

**【0033】**

次にステップS305にて、該データ量のカウント値が上記上限値をオーバーしたかどうかを検知し、オーバーしていなければステップS303のJPEG符号化処理を継続する。20これが初期状態の符号化フェーズである。

**【0034】**

符号化部102から出力する符号化データは、図4に示すように第1のメモリ104と第2のメモリ106の両方に格納されていく。縦縞で示した領域が該格納した符号を表現している。

**【0035】**

<<符号化・再符号化フェーズ>>

符号化部102の符号化処理が進行し、前記データ量のカウント値が設定されている上限値をオーバーすると、ステップS307にて、第1のメモリ104内の符号化データを廃棄すると共に、ステップS309にて、符号化部102の量子化ステップをQ2に変更する。30

**【0036】**

符号化データのデータ量のカウント値が設定された上限値をオーバーするという事は、圧縮後のデータ量が目標値以内に収まらないことを意味する。よって同じ量子化ステップを用いて符号化処理を継続しても意味が無いので、前よりもデータ量が少なくなるように、Q1よりも量子化ステップ幅の大きい量子化ステップQ2に変更するわけである。

**【0037】**

量子化ステップを変更した後、ステップS311では符号化部102の符号化処理を再開し、図5に示すように符号化データを第2のメモリ106のみに格納する。それと並行して、ステップS313の再符号化処理を行なう。再符号化処理では、第2のメモリ106に格納済みの符号化データを読み出して、再符号化部109にて再符号化処理を行ない、前記2つのメモリ104、106に格納する。そして、縦縞1の符号を全て再符号化するまで、該符号化処理と再符号化処理を継続する。再符号化部109から出力される再符号化データは、量子化ステップ変更後に符号化部102から出力される符号化データと同じ量子化ステップで符号化して得られる符号化データと全く同一の符号化データである。

**【0038】**

具体的にこの再符号化処理では、符号化データを一旦ハフマン復号した後の各量子化値に対して、これら値を $2^n$ で割った結果と同様の結果が出るビットシフト処理を施した後、50

再度ハフマン符号化を行なうことにより実現される。この方法は、ビットシフトのみで量子化ステップを変更する点と逆直交変換や再直交変換処理を行なわない点で、高速な再符号化処理が可能である。ステップ S 1 5 では、再符号化処理の終了検知が行なわれる。

#### 【 0 0 3 9 】

再符号化後のデータ量は再符号化前の符号化データのデータ量よりも少なくなるので、図 5 に示すように、再符号化前の符号を格納していたメモリ領域に再符号化後の符号化データを上書きするように格納することができる。再符号化処理が終了した時点で、縦縞 1 の符号化データのデータ量は図 6 に示すの斜め縞 1 の符号化データのデータ量へと減少する。

#### 【 0 0 4 0 】

以上で説明したステップ S 3 0 7 ~ S 3 1 5 が、符号化・再符号化フェーズで行なう処理である。

#### 【 0 0 4 1 】

<< 転送フェーズ >>

再符号化処理が終了したら、ステップ S 3 1 7 では転送処理が行なわれる。該転送処理では、図 6 に示すように、符号化・再符号化フェーズで第 2 のメモリ 1 0 6 のみに格納した斜め縞 2 の符号化データを、第 1 のメモリ 1 0 4 内の斜め線 1 の符号化データに連結されるアドレスに転送し、格納する。その一方で、第 2 のメモリ 1 0 6 上で分散してしまっている斜め縞 1 の符号化データと斜め縞 2 の符号化データが第 1 のメモリ 1 0 4 上で連続して格納される様に、前記斜め縞 2 の符号化データを第 2 のメモリ 1 0 6 内で転送し、連結させる。これが、転送フェーズで行なう処理である。

#### 【 0 0 4 2 】

上記転送フェーズが終了したら、ステップ S 3 0 3 、 S 3 0 5 の符号化フェーズに戻り、図 7 に示すように斜め縞 4 の符号を符号化部 1 0 2 から出力して 2 つのメモリ 1 0 4 , 1 0 6 に格納する。この符号化フェーズは、初期状態の符号化フェーズ（図 4 ）と少し異なり、符号化部 1 0 2 で符号化する際の量子化ステップが Q 1 から Q 2 に変更されていると共に、2 つのメモリ 1 0 4 , 1 0 6 に格納されている符号化データも様々なフェーズで処理された符号の集まりである。それらの違いを無視すれば、転送フェーズ直後の符号化フェーズと初期状態の符号化フェーズは、同じと見なせる。

#### 【 0 0 4 3 】

よって、符号化フェーズ、符号化・再符号化フェーズと転送フェーズの 3 つを繰り返すことで、最終的に 1 ページの画像データをデータ量設定値以下に圧縮した符号を第 1 のメモリに格納することが出来る。しかも、入力部 1 0 1 は一連の処理が終わるまで、入力を継続するだけである。すなわち、画像を再度最初から入力し直すということが無くなる。

#### 【 0 0 4 4 】

図 3 に示したフローチャートは、説明が理解しやすいように、図 4 、図 5 、及び、図 6 に示した各フェーズに対応する処理のみを記述した。しかしながら実際には、1 ページの画像データの入力はどこかのフェーズで終了する。従って、どのフェーズで終了したかによって、それ以降の対応も多少異なる。それを考慮した流れを示したのが図 8 のフローチャートである。図 8 のフローチャートは、1 ページ分の画像データの入力完了と図 3 で説明した各種処理との関係を考慮したものであり、ここでは図 3 のフローチャートに、ステップ S 8 0 1 、 S 8 0 3 、 S 8 0 5 、 S 8 0 7 を追加している。

#### 【 0 0 4 5 】

ステップ S 8 0 1 、 S 8 0 3 、 S 8 0 5 は、それぞれ、符号化フェーズ、符号化・再符号化フェーズ、転送フェーズにおいて、入力部 1 0 1 からの 1 ページ分の画像データの入力が終了したことを検知する。

#### 【 0 0 4 6 】

符号化フェーズと転送フェーズで 1 ページ分の画像データの入力が終了したことを検知した場合（ステップ S 8 0 1 、 S 8 0 5 ）、ステップ S 8 0 7 へ移り、当該ページの圧縮符号化処理を終了し、次に処理すべき 1 ページ以上の画像データがあれば、次の 1 ページ分

10

20

30

40

50

の画像データの圧縮符号化処理を開始し、無ければ停止状態に入る。

【0047】

一方、符号化・再符号化フェーズで1ページ分の画像データの入力終了を検知した場合(ステップS803)には、符号化部102では再符号化処理する画像データが無くなるまで一旦動作を止める必要があるので、ステップS311の符号化処理をパスし、ステップS313で、今までに符号化部102で符号化済みの画像データを所定の符号化データ量に抑える為の再符号化処理のみを継続して行なう。再符号化処理が全て終了して、その後の転送処理が終わらないと、1ページ分の画像データ全体の符号化データが第1のメモリ上に集まらないため、1ページ分の画像データの入力終了後も再符号化処理及びそれに続く転送処理は継続して行われる必要がある。この場合には、ステップS315にて、再符号化処理が全て終了したことを検知すると、符号化・再符号化フェーズ中に、第2のメモリ106のみに格納された符号化データを第1のメモリに転送し(ステップS317)た後、次のステップS805にて、1ページ分の画像データの入力終了が検知されてステップS807へ移ることになる。10

【0048】

以上が動作であり、図8の動作説明もある。

【0049】

<メモリ格納方法の変形例>

図9、図10は図5、図6の概念図で示したメモリ格納方法の変形例を示す図である。

【0050】

図5の概念図においては、符号化・再符号化フェーズでは、符号化部102から出力する符号化データは第2のメモリ106のみに格納していたが、図9に示すように符号化・再符号化フェーズ中に、符号化部102から出力する符号化データを第1、第2メモリの両方に直接格納する。20

【0051】

符号化部102から見ると、どのフェーズで符号化して出力する符号化データも両方のメモリへ格納することになる。また、図6の概念図とは異なり、図10に示す様に、転送フェーズでメモリ間のデータ転送が必要なくなる。またこの変形例の場合には、符号化・再符号化フェーズにおいて、符号化データと再符号化データを第1のメモリ104へ送った順序で順次格納される。そのため2種類のデータが入り混じってしまうという問題は有る30。

【0052】

従って、この変形例の場合にはこれに対応する為に符号化データをある単位で区切って、ファイル或いはパケットとして管理する様にする。具体的には、ファイル管理テーブル、或いは、パケット管理テーブル等を別に作成して管理する。

【0053】

一つの手法としては、符号化部102からのデータを第1メモリ104に格納する際、適当な単位(例えば前記直交変換の単位が $8 \times 8$ のブロックであるので、 $8 \times i$ ( $i = 1, 2 \dots$ の整数)ライン分のデータ)毎に、画像データの先頭から管理番号を割り当て、各管理番号に対応する符号化データの格納先頭アドレスと該符号化データ量とを、管理番号順に格納できるような管理テーブルを作成する。40

【0054】

符号化部102や再符号化部109は処理中のデータの管理番号を保持し、該管理番号に基づいて、符号化データ格納時の先頭アドレスと符号化データ量とを管理テーブルに書き込む。このようにすれば、符号化部102と再符号化部109で処理した符号化データをランダムに格納したとしても、前記管理テーブルを管理番号順にアクセスし、その時読み出させる先頭アドレスと符号化データ量に基づいて、符号化データを第1メモリ104から読み出せば、画像の先頭から順番に符号化データを読み出すことができる。このような管理機構を設ければ、画像上で連続するデータをメモリ上で連続するように格納する必要性が無くなる。50

**【 0 0 5 5 】**

図10の概念図における転送フェーズ後の符号化フェーズは、これまで説明した2つの符号化フェーズ（図4、図7）とほとんど同じであり、第1のメモリ内における符号の格納状態が図11に示した様に若干異なるだけである。よって、先の説明と本変形例は、3つのフェーズを繰り返して処理することに変わりはない。

**【 0 0 5 6 】**

次に、本発明において特徴的な符号化処理を行なう為の、第2の基本構成の例（これまで説明した構成を第1の例という）を図2を用いて説明する。

**【 0 0 5 7 】**

図2は、第2の例における画像処理装置200のブロック構成図である。

10

**【 0 0 5 8 】**

図1の画像処理装置100と大きく異なる点は、最初に符号化を行なう符号化部が2つ並列に存在する点である。画像処理装置200は、入力部201から入力される画像データを、第1の符号化部202と第2の符号化部205で並行して符号化し、互いに圧縮率の異なる2種類の符号化データを生成する。本例でも、符号化方式は公知のJPEG符号化方式を用い、 $8 \times 8$ 画素単位に相当する画像データを直交変換し、後述する量子化ステップを用いた量子化、ハフマン符号化処理を行なうものである。

**【 0 0 5 9 】**

なお、本例では第1の符号化部202よりも、第2の符号化部205の方が適用する圧縮率を高く設定する場合について説明する。具体的には、第1の符号化部202における量子化ステップをQ1、第2の符号化部205の量子化ステップをQ2（=  $2 \times Q1$ ）とする。

20

**【 0 0 6 0 】**

符号化部202から出力される符号化データは、第1のメモリ制御部203を経由して、第1のメモリ204に格納される。このとき、第1のカウンタ208は、符号化部202から出力される符号化データのデータ量をカウントし、これを保持すると共に、符号化シーケンス制御部209にも出力する。

**【 0 0 6 1 】**

一方、符号化部205で符号化された符号化データは、第2のメモリ制御部206を経由して、第2のメモリ207に格納される。このとき、第2のカウンタ210は、符号化部205から出力される符号化データのデータ量をカウントし、これを保持する。更に、後述する第2のメモリ207に格納している符号化データを第1のメモリ204に転送する時には、それと同時に上記カウント値を、第1のカウンタ208に転送する。

30

**【 0 0 6 2 】**

さて、第1のカウンタ208が符号化部202から出力される符号化データのデータ量をカウント中に、該カウント値がある設定値に達した時には、符号化シーケンス制御部209は、第1の例と同様、メモリ制御部203に対してメモリ204に格納されているデータを廃棄するよう制御信号を出す。

**【 0 0 6 3 】**

そして、符号化シーケンス制御部209は、第2のメモリ207に格納している符号化データを読み出して第1のメモリ204に転送し、第1のメモリ204に格納するよう、メモリ制御部206とメモリ制御部203に制御信号を出力する。この結果、第2のカウンタ210のカウント値が第1のカウンタ208に転送され、その値が第1のカウンタのカウント値としてロード（上書き）される。

40

**【 0 0 6 4 】**

要するに、上記第2のカウンタ210のカウント値は、第2のメモリ207に格納している符号化データのデータ量を表わしているので、そのカウント値と符号化データを、互いの対応付けが変わらない様に、そのまま第1のカウンタと第1のメモリへコピーしたと考えれば良い。

**【 0 0 6 5 】**

50

さらに、符号化シーケンス制御 209 は、第1の符号化部 202 および、第2の符号化部 205 に対して、今までよりも、符号化データが少なくなるような符号化を行なうように制御信号を出す。

#### 【0066】

例えば、第1の符号化部 202、及び、第2の符号化部 205 における量子化ステップ S を2倍に切り替えす。この結果、第1の符号化部 202 は、その直前までの第2の符号化部 205 における量子化ステップ Q2 (= 2 × Q1) を継承することになり、第2の符号化部 205 は更に大きな量子化ステップ Q2 × 2 を用いて、次のオーバーフローに備えた更に高い圧縮率の符号化処理を行うことになる。

#### 【0067】

ここでは、量子化ステップの倍率比を2倍としたがこれに限らず、任意に設定できることは示すまでもない。切り替えられた各符号化部 202、205 から出力された符号化データは、それぞれ、対応するメモリ制御部 203、206 を経由して、対応するメモリ 204、207 に格納される。

#### 【0068】

そして、符号化シーケンス制御 209 は、メモリ制御部 206 に対し、既に第2のメモリ内に格納している符号化データを読み出して、再符号化部 211 にデータを送るよう制御信号を出す。再符号化部 211 は、図1の再符号化部 109 と同様にして符号化データの再符号化処理を行なう。

#### 【0069】

第3のカウンタ 212 は、再符号化部 211 が出力したデータ量をカウントするもので、再符号化処理を開始する直前にゼロにリセットされ、再符号化処理中の出力データ量をカウントする。このカウンタ 212 は、再符号化処理が終了した時点で、そこで得られたカウント値を第2のカウンタ 210 に転送する。

#### 【0070】

第2のカウンタ 210 は、上記転送されてきたデータ量カウント値を、第2のカウンタ 210 内に保持しているカウンタ値に加算することにより、再符号化処理中にメモリ 207 に格納した、符号化データと再符号化データの合計のデータ量を算出する。即ち、メモリ 207 に格納しているデータ量とカウンタ 210 のカウント値とが一致する。

#### 【0071】

再符号化処理の終了 / 未終了に関わらず、符号化するべき入力部 201 からの画像データが残っていれば、2つの符号化部 202 と 205 による符号化処理を継続して行なう。そして、カウンタ 208 のカウント値がある設定値に達したかどうかの監視は入力部 201 から入力される1ページ分の画像データの符号化処理（符号化、再符号化）が終わるまで繰り返され、上述した符号化と再符号化の処理は、ここで得られる検出結果に応じた制御の上で実行される。

#### 【0072】

上記図2の構成における処理のフローを表わすフローチャートを図12に示す。

#### 【0073】

図2で説明したように符号化部が2つある場合は、図12に示すフローチャートに基づいて1ページ分の画像データの符号化を行なう。なお、図12の説明は、符号化部が1つの場合のフローチャートである図8とは、大半は類似しており、当業者であれば上記説明から本第2の例の特徴は十分に理解できるであろうから、符号化部1つの場合と同じように3つのフェーズで処理を説明する様にし、図8と異なる点を主に説明することとする。

#### 【0074】

上述した図8のフローと本例のフローとの一番大きな違いは、ステップ S317 の転送処理が、ステップ S307 とステップ S309 の間に移動していることである。要するに、符号化・再符号化フェーズと転送フェーズが入れ替わったと見なせば良い（ステップ S307 の符号化データの廃棄処理は例外である）。

#### 【0075】

10

20

30

40

50

ステップ S 3 0 1 の符号化パラメータの初期設定では、第 1 の符号化部 2 0 2 に量子化ステップ Q 1 を、第 2 の符号化部 2 0 5 には量子化ステップ Q 2 (= 2 × Q 1) を設定する。

#### 【 0 0 7 6 】

符号化フェーズでは、ステップ S 8 0 1、S 3 0 3、S 3 0 5 を繰り返し実行する。ステップ S 8 0 1 とステップ S 3 0 5 は符号化部が 1 つの場合と同じ処理であるが、ステップ S 3 0 3 の符号化処理だけは図 1 3 に示すように異なっている。

#### 【 0 0 7 7 】

第 1 のメモリ 2 0 4 へ格納する符号化データは圧縮率が段階的に高くなるようにするため、最初に格納する符号化データは圧縮率が一番低い量子化ステップ Q 1 で符号化したデータを格納し、第 2 のメモリへ格納する符号化データは量子化ステップ Q 2 で符号化したデータを格納する。

#### 【 0 0 7 8 】

第 1 のメモリ 2 0 4 へ格納中のデータ量が設定されている上限値をオーバーしたら（ステップ S 3 0 5）、直ちに、第 1 のメモリ 2 0 4 で保持していた符号化データを廃棄し（ステップ S 3 0 7）、第 2 のメモリ 2 0 7 で保持している圧縮率の高い符号化データを、第 1 のメモリ 2 0 4 へ転送する（ステップ S 3 1 7、図 1 4 参照）。これにより、第 1 の例（図 1）で説明した 1 回目の再符号化処理の終了を待たずに、速やかに、上限値をオーバーしない適切な 2 番目の候補の符号化データを第 1 のメモリ 2 0 7 内に格納出来る。これが、図 1 に対する、2 つの符号器を持つ図 2 を適用することの最大の利点である。

#### 【 0 0 7 9 】

本第 2 の例では、2 つのメモリ 2 0 4、2 0 7 で同じ圧縮率の符号化データを持っていることが無駄という考え方なので、第 2 のメモリ 2 0 7 には、第 1 のメモリ 2 0 4 に格納する符号化データよりも圧縮率の高い符号化データを格納しておくようになっている。従って、それ以降の処理もこの考え方に基づき行われるものであり、第 2 のメモリ 2 0 7 内の符号化データを第 1 のメモリ 2 0 4 に転送する処理（転送フェーズ）が終了した後は、第 2 のメモリ 2 0 7 の符号化データを、更に 1 段階圧縮率の高い符号化データを保持する様に再符号化することとなる。

#### 【 0 0 8 0 】

具体的には、まず図 1 5 に示す様に、転送フェーズの次の符号化・再符号化フェーズでは、上記再符号化の前に、2 つの符号化部 2 0 2、2 0 5 に適用される各量子化ステップ Q 1、Q 2 をそれぞれ Q 2、Q 3 へ変更し（ステップ S 3 0 9）、1 ページの画像データの入力が終了せずに続いていれば（ステップ S 8 0 3）、後続の画像データは新たな量子化ステップが設定された 2 つの符号化部で該入力データを符号化して（ステップ S 3 1 1）、対応する各メモリ 2 0 4、2 0 7 へ格納する。そして、上記符号化処理と並行して第 2 のメモリに格納されている符号化データ（第 1 のメモリ 2 0 4 に転送したもの）は、第 1 のメモリ内の符号化データよりも 1 段階高い圧縮率の符号化データに変更するべく、再符号化部 2 1 1 にて量子化ステップ Q 3 を用いて符号化されたデータが得られる様な再符号化処理（S 3 1 3）を行ない、再符号化データを第 2 のメモリ 2 0 7 に格納し直す。

#### 【 0 0 8 1 】

なお、本第 2 の例でも、第 1 の例と同様、再符号化処理では、符号化データを一旦ハフマン復号した後の各量子化値に対して、これら値を  $2^n$  で割った結果と同様の結果が出るビットシフト処理を施した後、再度ハフマン符号化を行なうことにより実現される。この方法は、ビットシフトのみで量子化ステップを変更する点と逆直交変換や再直交変換処理を行わない点で、高速な再符号化処理が可能である。

#### 【 0 0 8 2 】

なお、本第 2 の例の様に符号化部が 2 つ有る場合には、図 1 5 に示したように、第 2 のメモリ 2 0 7 に符号化データと再符号化データを混在して格納する状況が発生する。従って、前述したように、符号化データのある単位で区切って、ファイル或いはパケットとして管理することが、第 2 のメモリ 2 0 7 に対しても必要になる。その為には、例えば第 1 の

10

20

30

40

50

例における変形例と同様の構成を設ければ良いであろう。

#### 【0083】

図12において、再符号化処理の終了をステップS315で検知したら、また符号化フェーズ（ステップS801、S303）に移行する。なお、符号化・再符号化フェーズ後の符号化フェーズでは、図16に示すように、2つのメモリ204, 207が保持する符号化データは圧縮率が違うだけでなく、符号化データの混在の仕方（アドレス）もかなり違ってくる。従って、再度、第1のメモリ204のデータ量が設定値をオーバーした場合には、第2のメモリ207で保持されている符号化データ（ $6 + 8$  の横縞の領域の符号）が第1のメモリ204へ転送される必要が出てくる。これらを考慮すると、第2のメモリ207だけでなく、第1のメモリ204でも符号化データをファイル或いはパケットとして管理する必要がある。よって、第1のメモリ204にも前述の管理テーブルを用いた管理機構が必要となる。10

#### 【0084】

図16に示された符号化フェーズの状態は、量子化ステップと符号化データの混在の仕方が、再符号化処理の前後で異なっていること以外は、初期状態の符号化フェーズ（図13）と同じである。よって、符号化フェーズ、転送フェーズと符号化・再符号化フェーズを繰り返すことで、最終的に、1ページ分の画像データを設定した上限値以下に圧縮した符号化データを確実に第1のメモリ204に格納することが出来る。

#### 【0085】

なお、第1の例の説明とは、転送フェーズと符号化・再符号化フェーズの配置順が逆であることから、図8において転送処理後に行なっていた1ページ分の画像データの入力終了検知（ステップS805）は、符号化・再符号化フェーズで行なう1ページ分の画像データの入力終了検知（ステップS803）と、ほとんど同じタイミングになってしまふ。また、2つの検知処理は、機能的にはステップS805と同じで、タイミング的にはステップS803と同じである。従って、これら2つのステップは、新たな1ページ分の画像データの入力終了を検知するステップとして統合し、ステップS1201と表記しておく。20

#### 【0086】

以上説明した第1、第2の例では、第1のメモリと第2のメモリは物理的に別のメモリであるとして説明をしてきた。これは、2つのメモリに対するアクセスが独立したものとすることができるので有利なためであり、本発明の特徴となす。しかしながら、第1のメモリと第2のメモリを、物理的に別のメモリとしない場合も本発明の範疇に含まれる。物理的に1つのメモリ上に、前記第1のメモリと第2のメモリに相当する2つの領域を確保して、第1のメモリを第1のメモリ領域、第2のメモリを第2のメモリ領域と言い直して、これまでの説明を読み直せば、本発明は、1つのメモリでも実現できることが分かる。30

#### 【0087】

また、1つのメモリで上記各例を実現する場合には、前記転送フェーズで説明したデータ転送処理のいくつかは不要となる。その詳細はその都度容易に想像できるので説明は省略するが、前記2つの領域を厳密に別けて使用する場合、物理的に2つのメモリを持つ時と同じようにデータ転送処理が必要であるが、2つの領域間で同じデータを共有することになれば、データ転送処理が不要になるだけでなく記憶容量の削減も図れる。40

#### 【0088】

例えば、第2のメモリ領域で保持していた符号化データを、第1のメモリ領域へ転送する際、該符号化データが格納されている先頭アドレスとデータサイズの2つの情報を第2のメモリ制御部から第1のメモリ制御部へ転送するだけで、前記符号化データを転送したのと同じ効果が得られる。

#### 【0089】

前記符号化データを、ファイル形式やパケット形式で格納している場合は、メモリ制御部の間で転送する情報は少し増え、該符号化データに関連する管理テーブル情報を転送する必要がある。それでも、符号化データを転送するよりは、効率が良い。

#### 【0090】

さて、上述した画像処理装置によると、入力した画像データを符号化していく際に、目的とするサイズに越えるような場合であっても、その入力を継続しつつ目標とするサイズに収めるよう処理を継続することができるようになるが、本発明では、上記画像データの圧縮データの符号量の制御以外に、該画像データに付随する像域情報を符号化したデータの符号量の制御も行なうものである。

#### 【0091】

以下に説明する実施形態では、像域情報の符号化処理と符号化データの符号量の制御を行なうかを説明する。

#### 【0092】

<第1の実施形態>

上述の画像処理装置に本発明を適用した第1の実施形態を図17に示す。同図は図1に示した基本構成に本発明を適用したもので、図1の構成と同じ機能ブロックには、同一番号を付することで、その説明は省略する。

#### 【0093】

イメージスキャナやページ記述言語レンダリングなどから入力部101を通して入力した画像データは、既に説明した処理方法に基づいて符号化と再符号化処理を繰り返して行ない、符号化データを設定した符号量以内に収める。

#### 【0094】

一方で、画像データは像域情報生成部1701に供給され、前述の像域情報を生成する。本実施形態では、スキャナ入力画像について説明しているので、画像データのみに基づいて像域情報を生成するが、ページ記述言語(PDL)を展開・描画した画像の場合、該PDL情報を参照して像域情報を生成すれば良い。なお、像域情報はスキャナ等の画像入力機器で生成することもある。その場合、前記像域情報も入力部101を通して入力され、像域情報生成部1701を素通りして、次のユニットへ送られる。

#### 【0095】

さて、像域情報生成部1701で生成する像域情報であるが、実施形態では、着目画素が文字・線画領域であるか中間調領域であるのか、及び、有彩色か無彩色なのかを示す情報を生成する。それぞれ1ビットで表現できるので、1画素について合計2ビットの像域情報(それぞれのビットは、像域情報を構成することになるので像域成分情報という)を生成することになる。

#### 【0096】

像域情報生成部1701の動作を簡単に説明すると、次の通りである。

#### 【0097】

先ず、文字・線画領域か中間調領域であるのかの判断であるが、文字・線画の場合は背景に対してその輝度(或いは濃度)が急峻に変化する。一方、中間調領域にある場合、注目画素の輝度(或いは濃度)は隣接する画素に対して変化が小さい。従って、注目画素の輝度を $L_i$ 、左右の画素の輝度を $L_{i-1}$ 、 $L_{i+1}$ と定義したとき、

$$|L_i - L_{i-1}| > T$$

又は、

$$|L_i - L_{i+1}| > T$$

である場合に、注目画素は文字線画(のエッジ)にあると判断できる。尚、ここでの $|x|$ は $x$ の絶対値を示すものである。

#### 【0098】

なお、濃度が急峻に変化しているか否かを判断する式は上記のものに限らない。例えば、

$$|2L_i - L_{i-1} - L_{i+1}| > T$$

を満たすかどうかで判断してもよいし、一次元方向のみではなく、2次元方向について判断するようにしても良い(但し、2次元で判断する場合には、像域情報生成部1701内には、複数ライン分の画像データを記憶するためのメモリを必要とする)。

#### 【0099】

一方、有彩色/無彩色の判断であるが、入力される画像データはスキャナより読み取られ

10

20

30

40

50

たものであるので、R、G、Bのデータ形式になっている。

#### 【0100】

無彩色というのは、RGBの各色成分が互いに同じ輝度である場合であるので、

$$R = G = B$$

なる関係を有する場合には無彩色として判断し、それを満たさない場合には有彩色と判断する。ただし、スキャナ装置が有するCCDの精度も加味する必要があるので、実際は、

$$B - < R < B +$$

$$R - < G < R +$$

$$G - < B < G +$$

の全てを満たすとき無彩色と判断し( は適当な小さい数値 )、それ以外を有彩色と判断するようにしても良い。

#### 【0101】

場合によっては、RGB色空間を例えば輝度、色相、彩度( 例えばLab表色空間 )に変換し、その彩度が所定値以下の場合に無彩色、所定値を越える場合に有彩色と判断しても良いであろう。

#### 【0102】

以上の如く、像域情報生成部1701は、入力された画像データから注目画素が文字・線画 / 中間調領域か、有彩色か無彩色かを示す2ビットの像域情報を生成し、出力する。

#### 【0103】

図17の説明に戻る。上記のようにして、生成した像域情報は、ブロック化ユニット1703にて、まとめて符号化するデータのサイズ、例えばM×Nのサイズ( 従ってM×N×2ビット )へブロック化する。ブロックサイズは、32×32とするが、符号化効率を上げるため、それ以上、例えば64×64や128×128でも構わないし、必ずしも正方サイズである必要もない。それ故、M×Nサイズとした。

#### 【0104】

ところで、像域情報を符号化する場合、通常の画像データの圧縮に利用するJPEGのような多値の非可逆圧縮は適さない。それ故、可逆圧縮であるJBIG、或いはPacKBits等のランレンゲス符号化を用いることとした。可逆符号化部1705では、該当するブロックの像域情報を、この可逆符号化する。

#### 【0105】

符号化した像域情報は、第3のメモリ制御部1707を経由して、第3のメモリ1709に格納する。また、これと同時に、前記可逆符号化部1705から出力する符号量を第4のカウンタ1711が累積カウントしており( 1ページの読み取りを開始する際にリセットされる )、その結果を符号化制御部1713に供給している。

#### 【0106】

符号化制御部1713には、不図示のレジスタが設けられ、このレジスタに像域情報の目標符号量が予め設定されている。そして、符号化制御部1713は、第4のカウンタ1711で計数されている符号量が、この目標値をオーバーしたか否かを監視している。そして、目標値をオーバーしたと判断した場合、符号化制御部1713は、可逆符号化部1705及び可逆符号再符号化部1707にそれぞれに対して以下に示す指令を発すると共に、第4カウンタ1711をリセットする。

#### 【0107】

先ず、可逆符号再符号部1707に対しては、第3のメモリ1709から符号化した像域データを読み出し再符号化するよう指令を発する。この結果、可逆符号再符号化部1715は、第3のメモリ1709から符号化されたブロック単位の像域情報を読み出し、それを一旦復号し、ランレンゲスを高める処理を行い、再度符号化して、第3のメモリ1709に格納する。

#### 【0108】

このランレンゲスを高める処理であるが、次の2通りを用意した。

処理P1：文字・線画 / 中間調画素の識別ビットが中間調画素を示している画素位置の、

10

20

30

40

50

有彩色 / 無彩色の識別情報を全て有彩色に変更する。

処理 P 2 : 文字・線画 / 中間調画素の識別ビットを中間調画素を示すように変更する。

#### 【 0 1 0 9 】

符号化制御部 1 7 1 3 は、或るページの像域情報を符号化している最中に、最初に目標値をオーバーすると判断した場合には、上記の処理 P 1 を採用する。そして、この処を行っても、目標値をオーバーすると判断した場合（2 回目にオーバーすると判断した場合）には、処理 P 2 を用いて再度符号するようにした。

#### 【 0 1 1 0 】

有彩色 / 無彩色の識別情報を有彩色にすることは、色空間において有彩色空間が無彩色を含んでいるため、大きな問題にはならないからである。また、文字・線画 / 中間調の識別情報 10 を中間調にすることも、同様の理由による。

#### 【 0 1 1 1 】

いずれにしても、かかる像域情報を変更することで、そのエントロピーが低下するため、ランレンジング符号化後のデータ量は減少する。再符号化後の属性データは、メモリ 1 7 0 9 に再び格納すると共に、その符号量を第 5 のカウンタ 1 7 1 7 にてカウントする。そして、可逆符号再符号化部 1 7 1 5 は、可逆符号化部 1 7 0 5 によって既に符号化した全符号についてこの処理を行っていく。こうして、符号化制御部 1 7 1 3 から指令があった時点で、第 3 のメモリ 1 7 0 9 に格納されていた符号データ（可逆符号化部 1 7 0 5 で符号化されたデータ）に対する再符号化及び格納を行う。このときの符号量は、第 5 のカウンタ 1 7 1 5 で計数しており、その計数結果が第 4 のカウンタに加算される。

#### 【 0 1 1 2 】

一方、可逆符号化部 1 7 0 5 は、符号化符号化制御部 1 7 1 3 から目標値をオーバーした場合に発した命令に従い、注目ブロック以降について、上記の可逆符号再符号化部 1 7 1 5 と同じ処理を開始する。すなわち、或るページの符号化を行って最初にオーバーすると判断された場合には、先に示した処理 P 1 でもってブロック化部 1 7 0 3 からのブロックについてランレンジングを高める処理を行い、その結果を符号化して第 3 のメモリ 1 7 0 9 に格納する。この間、可逆符号化部 1 7 0 5 で符号化された符号量は、第 4 カウンタ（オーバーすると判断された時点でリセットされる）で計数回数を開始するが、先に説明したように、可逆符号再符号化部 1 7 1 5 でも再符号化が行われた際の、符号量（第 5 のカウンタ 1 7 1 7 ）と加算されることになるので、結果的に、1 ページについての属性情報について、処理 P 1 で行った結果と同様の符号量を第 4 カウンタ 1 7 1 1 が計数することになる。

#### 【 0 1 1 3 】

こうして、処理 P 1 に切り換わって像域情報の符号が行われている最中に、再度目標値をオーバーすると判断した場合、符号化制御部 1 7 1 3 は、第 4 カウンタ 1 7 1 1 をリセットすると共に、可逆符号化部 1 7 0 5 及び可逆符号再符号化部 1 7 1 5 に対して、先に示した処理 P 2 に切り換わるよう指令を発する。

#### 【 0 1 1 4 】

以上をまとめると、次のようになる。

#### 【 0 1 1 5 】

符号化制御部 1 7 1 3 は、可逆符号化部 1 7 0 5 における像域情報の符号化量を監視していく、その符号量が目標値をオーバーすると判断した場合、可逆符号化部 1 7 0 5 及び可逆符号再符号化部 1 7 1 5 に対して処理 P 1 で行うように設定し、第 4 カウンタ 1 7 1 1 をリセットする。第 4 カウンタ 1 7 1 1 は、これ以後、可逆符号化部 1 7 0 5 による符号量及び第 5 カウンタ 1 7 1 7 のカウント値を加算していく。従って、可逆符号再符号化部 1 7 1 5 による再符号化が完了した時点では、1 つの画像（或いは 1 ページの画像）について最初から処理 P 1 で行ったときと実質的に同じ符号量を計数することとなる。また、第 3 のメモリ 1 7 0 9 には処理 P 1 での符号化データが格納されることになる。

#### 【 0 1 1 6 】

この処理 P 1 による処理中に、再度、目標値をオーバーすると判断した場合、今度は、可

10

20

30

40

50

逆符号化部 1705 及び可逆符号再符号化部 1715 に対して処理 P2 で行うように設定し、第4カウンタ 1711 及び第5カウンタ 1717 をリセットするようとする。

#### 【0117】

以上の結果、入力部 101 からの画像データの入力を継続しつつ、目標値（目標サイズ）以内に像域情報を可逆符号化させることができるようになる。

#### 【0118】

次に、本第1の実施形態の処理内容を表わすフローチャートを図18に示し、該フローチャートを用いて説明を行なう。本実施形態の処理は、上述したように、大きく2つの処理に分かれる。1つは可逆符号化処理、もう1つは再符号化処理である。

#### 【0119】

符号化処理は、ステップ S1805 の像域情報変換処理とステップ S1807 の可逆符号化処理とからなり、再符号化処理は、ステップ S1815 の復号化処理と、ステップ S1817 の像域情報変換処理、それにステップ S1819 の再符号化処理とからなる。

#### 【0120】

その他のステップは、S1813 の像域情報変更処理の変更を除くと、条件分岐と開始及び終了ステップである。

#### 【0121】

まず、ステップ S1801 から、像域情報の符号化処理を開始する。次のステップ S1803 で、画像1ページに対する像域情報の入力が全て終了しているかどうかを検知し、終了していればステップ S1821 へ行き、終了していない場合、すなわち、像域情報の入力がある場合には、ステップ S1805 の像域情報変換処理を行なう。

#### 【0122】

再符号化処理がまだ一度も起動していない初期状態では、像域変換処理では何も処理せず、入力された像域情報をそのまま次のステップ S1807 の可逆符号化処理で符号化する。

#### 【0123】

次のステップ S1809 では、再符号化処理中であるかどうかを判定し、再符号化処理中であれば、前記ステップ S1815 ~ S1819 から成る再符号化処理を行なうが、再符号化処理中でなければ、ステップ S1811 へ行き、符号化したデータ（符号量）が設定値をオーバーしたかどうかを判定する。

#### 【0124】

オーバーしていなければ、ステップ S1803 に戻り、符号化処理を繰り返し、オーバーしていれば、像域情報変換処理における処理内容を変更する。該変更により、ステップ S1805、S1817 における像域変換処理の内容が変わる。具体的には、初期状態では、処理 P1、すなわち、入力された像域情報中の、文字・線画 / 中間調の識別情報が中間調であると判断された画素位置の、有彩色 / 無彩色の識別情報を全て有彩色とする。

#### 【0125】

この像域情報変換処理内容の変更後は、再符号化処理を行なう。再符号化処理では、ステップ S1815 にて、符号化した像域情報を復号化処理して符号化前のデータに戻す。次のステップ S1817 にて、前記像域情報変換処理を行ない、上記像域情報変更処理を行い、次のステップ S1819 にて、変換した像域情報を再び可逆符号化する。

#### 【0126】

上記再符号化処理が終わったら、ステップ S1803 に戻り、像域情報の入力があれば、それを符号化処理する。符号量が一度でも設定値をオーバーすると、像域情報変換処理内容が変更され、ステップ S1805 の像域情報変換処理にて、処理 P1 に置き換えられた後に、ステップ S1807 にて、可逆符号化される。

#### 【0127】

こうして、処理 P1 に置き換わった後に、再び、ステップ S1811 で符号量が目標値をオーバーしたと判断した場合には、ステップ S1813 で、ステップ S1805 及び S1817 における像域変更を処理 P2 に変更するように設定する。

10

20

30

40

50

**【 0 1 2 8 】**

いずれにしても、像域情報の入力が全て終了し、且つ、再符号化処理もしていなければ、ステップ S 1 8 2 3 へ移り、像域情報の符号化処理を終了する。

**【 0 1 2 9 】**

以上説明したように、本第 1 の実施形態によれば、1 つ、或いは 1 ページの画像データの入力を中断したり、再入力することなく、目標サイズ以内に画像データを符号化させると共に、その画像の像域情報についても目標サイズ以内に可逆符号化させることができるようになる。

**【 0 1 3 0 】**

実施形態における可逆符号化部及び可逆符号再符号化部が p a c k b i t s 符号化を採用した場合の例を以下に説明する。 10

**【 0 1 3 1 】**

ここでは、1 画素あたり 2 ビットの像域情報に、6 ビットデータ “ 0 0 0 0 0 0 ” を付加して 8 ビット化した後、可逆符号として P a c k b i t s 符号化を行なった場合の、本実施形態における具体的な処理内容を図 1 9 を用いて説明する。

**【 0 1 3 2 】**

P a c k b i t s 符号化する前の 8 ビットデータは、図 1 9 ( a ) に示すように上位 6 ビットは全て 0 で、下位 2 ビットの上位側に、対応する画素データが文字・線画 / 中間調を識別するためのフラグ、下位側には有彩色 / 無彩色を表わすフラグが格納されているものとする。よって、該 8 ビットデータが取り得る値は、0 以上 3 以下の値である。また、説明を簡単にするため、像域情報生成部 1 7 0 1 からは 1 次元の像域情報が出力されるものとして説明する。 20

**【 0 1 3 3 】**

像域情報生成部 1 7 0 1 からは、上記 8 ビットのデータが画素単位で出力される。具体的な出力データとして、図 1 9 ( b ) に示すデータを考える。

**【 0 1 3 4 】**

これを P a c k b i t s 符号化すると図 1 9 ( c ) に示すデータに圧縮される。圧縮後のデータで負の値は連続するデータの個数を表わし、非連続データの個数は正の値で表わしている。これらは長さ情報と言うもので、該長さ情報のサインビットから連続データが続くのか、非連続データが続くのかを判別することが出来るようになっている。圧縮後の各データは ( b ) と同じく 8 ビット ( 1 バイト ) である。1 バイトの長さ情報で、表わすことができる最大値は 2 5 5 の半分の約 1 2 8 であり、長さ情報がそれ以下の場合は 1 組の長さ情報とそれに続く像域フラグデータ群で符号化でき、それを超える場合は複数の組の、長さ情報 + 像域フラグデータ群、に分けて符号化される。 30

**【 0 1 3 5 】**

図 1 9 ( c ) の圧縮データを詳しく見てみることにする。最初の長さ情報「 - 4 」はマイナスの値なので、上述したように連続データの連続個数を表わし、長さ情報直後の像域フラグデータ「 1 」が 4 つ続くことを表わしている。

**【 0 1 3 6 】**

次のデータ「 4 」はまた長さ情報であるが、今度はプラスの値なので非連続データが 4 つ続くことを示している。よって、前記「 4 」に続く 4 つのデータ「 2 , 3 , 2 , 3 」が非連続データを表わす。図 1 9 ( c ) では、長さ情報と像域フラグデータとが区別し易いように、プラスの長さ情報のみ下線を引いている。 40

**【 0 1 3 7 】**

上記非連続データの次の「 - 5 」は又、連続データの長さ情報で、該長さ情報直後の像域フラグデータ「 2 」が 5 つ続くことを表わしている。次の下線付きのデータ「 3 」は非連続データの長さ情報で、後続する 3 つのデータ「 1 , 0 , 1 」が像域フラグデータであり、さらに次の「 - 6 , 0 」は、データ「 0 」が 6 個連続することを示している。

**【 0 1 3 8 】**

上記圧縮データを可逆符号再符号化部 1 7 1 5 で再符号化処理するとどのようになるのか 50

を、図19(d)(e)を用いて説明する。ここで説明を簡単にするため、再符号化処理において、有彩色・無彩色フラグのビットを「1」に固定することで、すべて有彩色にしてしまう場合について説明する。

#### 【0139】

符号化された像域データは一旦復号され、図19(b)のデータに戻された後、上記フラグデータの置き換えが行なわれることで、図19(d)のデータに変換される。これを再び、P a c k b i t s 符号化すると、同図(e)の符号化データが得られる。再符号化前は15バイトの符号化データであったのに対し、再符号化後には6バイトにまで減少することが解かる。

#### 【0140】

上記の再符号化処理を行なったにもかかわらず、全符号量の計数値が符号化制御部1713内のレジスタに設定された目標値を再びオーバーした場合には、前記再符号化処理が終了していれば、直ちに次の新たな再符号化処理を開始する。前記再符号化処理が終了していないければ、該再符号化処理が終了後、直ちに次の新たな再符号化処理を開始する。

#### 【0141】

新たな再符号化処理では、残りの1ビットの像域フラグも“1”に置き換える。これにより、すべての像域フラグデータ(8ビット)の値は“3”となり、データのバイト数をNとすると、符号化後のデータ量はおよそ( $2N / 128$ ) + 2バイトとなる。

#### 【0142】

これは、連続データの個数が128個を超えるたびに、あらたな2バイト1組の符号化データ(長さ情報と連続データ)が増えるためである。

#### 【0143】

P a c k b i t s の符号化回路や復号回路それにデータ変換回路はそれぞれ公知の技術であるため、個別の回路構成についての説明は省略する。

#### 【0144】

なお、上記実施形態では、説明では簡単化のため、各画素の像域フラグを2ビットとして説明したが、前述したように像域フラグとして他の情報を付加させることによってよい。

#### 【0145】

理論上、像域情報を変換する処理は、像域の種別としてNビットであれば $2^N$ 個存在することになる。但し、変換する毎に、圧縮率が高くなる必要がある。いずれにしても、像域情報のビット数が多ければ多いほど再符号化処理の回数を増やすことができ、符号量を多段階で制御することができる。

#### 【0146】

上述したように像域情報データの可逆符号化処理は、画像データの圧縮符号化処理とは独立に制御され、それぞれに目標符号量以内のデータに収められる。

#### 【0147】

符号化された2種類のデータは、外部に接続されるネットワーク機器、画像出力装置や大容量記憶装置等へ出力する際に多重化する。該多重化を考慮して、前記2種類のデータを符号化処理する単位を同じサイズに合わせておき、1単位を符号化して生成される符号化データを1つのパケットあるいはファイルとして管理・格納する。多重化する際に、画像位置が同じ2種類のパケットデータを、例えば画像データ・像域データの順に連結して、1パケット化し、外部へ出力する。

#### 【0148】

前記2つの符号化処理は独立に制御されるので、画像データの圧縮符号化処理部が他の構成であっても構わない。よって、像域情報データを可逆符号化処理するユニット1701～1717を前記図2の構成に対して付加しても、同じように処理することができる。その構成を図20に示しておくが、ユニット1701～1717の動作は図17とまったく同じであり、画像データ圧縮部分の動作も図2の動作とまったく同じである。

#### 【0149】

<第1の実施形態の変形例>

上記実施形態では、像域成分情報の 2 ビットは、一方が文字・線画 / 中間調の識別ビット、他方が有彩色 / 無彩色のビットとし、それらを 1 つの値として 0 ~ 3 の値を持つものとして扱った。しかしながら、文字・線画 / 中間調の識別情報と、有彩色 / 無彩色の識別情報は、本来は独立した情報である。そこで、文字・線画 / 中間調の識別ビットで構成されるブロックと、有彩色 / 無彩色の識別ビットで構成されるブロックを独立させ、それぞれ独立して符号化するようにしても良いであろう。この結果、それぞれのブロック内の各ビットは同じ種類のビットのみで構成されることになり、最初の段階で圧縮効率を高くできるようになり、その結果、像域情報の一部のビットを固定値に変換する確率を低くして、原像域情報のまま圧縮符号化できる確率を高くできる。

## 【0150】

10

## &lt; 第 2 の実施形態 &gt;

第 2 の実施形態では、像域情報のビット数が前記第 1 の実施形態と同じでも、再符号化処理の回数を増やせて、符号量を細かく制御することが出来る処理方法について説明する。本実施形態の構成も前記第 1 の実施形態と同じであり、違うのは、可逆符号化部 1705 と可逆符号再符号化部 1715 における像域フラグデータの圧縮方法である。

## 【0151】

前記第 1 の実施形態では、像域フラグデータを 1 ビットずつ固定値に置き換えるものであったが、本第 2 の実施形態では、状態数を縮退させる。例えば、2 ビットの像域フラグでは、4 状態を表わすことができるが、これを 1 回目の再符号化処理で 3 状態に縮退させ、2 回目の再符号化処理で 2 状態に縮退させることで、符号化前の情報エントロピーを減らし、符号化後のデータ量（符号量）を細かく減らしていくものである。

20

## 【0152】

状態数という言葉を用いて、前記第 1 の実施形態を表現すると、像域情報を再符号化するごとに状態数を半分に減らす、と言うことが出来る。

## 【0153】

1 回の再符号化処理で、上記第 1 の実施形態では実質的に状態数を半分に減らすのに対し、本実施形態は状態数を 1 つずつ減らすわけであるから、符号量を細かく減らせるのは当然である。

## 【0154】

本第 2 の実施形態の処理結果を図 21 (a) 乃至 (e) を用いて説明する。

30

## 【0155】

同図 (a) は、図 19 (b) に示した像域フラグデータと同じ、2 ビットの 4 状態の全てが存在するデータである。該 4 状態の定義は、次のようにする。

(1) 有彩色の文字線画部（データ「3」に対応）

(2) 無彩色の文字線画部（データ「2」に対応）

(3) 有彩色の中間調（データ「1」に対応、有彩色の画像部とも言う）

(4) 無彩色の中間調（データ「0」に対応、無彩色の画像部とも言う）

本第 2 の実施形態では、1 回目の再符号化処理で、上記 4 状態の内、(3) 有彩色の中間調と、(4) 無彩色の中間調の 2 状態を合体して 1 つの状態 (3') 中間調に縮退させる。これにより、以下の 3 つの状態となる。これは、前記処理 P1 に対応する。

40

(1) 有彩色の文字部

(2) 無彩色の文字部、

(3') 中間調

具体的には、データ「0」を「1」に置き換えることで、上記状態の縮退を行なう。状態縮退後のデータは図 21 (b) に示すデータに変わる。これを Packbits 符号化すると図 21 (c) に示す符号化データになる。再符号化前の符号化データ図 19 (c) よりも多少符号量が少なくなっていることが解かる。

## 【0156】

2 回目の再符号化処理では、(1) 有彩色の文字部と (2) 無彩色の文字部、の 2 状態を合体して 1 つの状態 (1') 文字部に縮退させる。これにより、以下の 2 状態となる。

50

( 1' ) 文字部

( 3' ) 非文字部

今度は、データ「 2 」を「 3 」に置き換えることで、上記状態の縮退を行なう。状態縮退後のデータは図 21 ( d ) に示すデータに変わる。このデータは図 19 ( d ) のデータと同じである。これを P a c k b i t s 符号化した符号化データ図 21 ( e ) は当然図 19 ( e ) と同じ符号化データになる。

#### 【 0 1 5 7 】

第 1 の実施形態では、再符号化前の 15 バイトのデータが 1 回の再符号化で 6 バイトまで減少したが、本実施形態では、1 回目の再符号化で 11 バイトに減少し、2 回目の再符号化でやっと 6 バイトに減少する。結局、第 2 の実施形態では、第 1 の実施形態では得られない細かく変化して減少する符号量を得ることが出来るので、目標符号量に近い像域フラグの圧縮データを得ることができる。10

#### 【 0 1 5 8 】

< 適用例 >

上記第 1 の実施形態及びその変形例、更には第 2 の実施形態では、イメージスキャナから画像を読み取る装置を例にし、その装置の機能動作について説明した。そして、その機能のほとんど（符号化処理も含む）は、上述した様にコンピュータプログラムによって実現できる。

#### 【 0 1 5 9 】

従って、本発明はパーソナルコンピュータ等の汎用情報処理装置上で動作するアプリケーションプログラムに適用しても構わない。アプリケーションプログラムに適用する場合には、圧縮元となる画像ファイルをユーザに指定させると共に、目標サイズをユーザに選択させる等の G U I を設ければ良いであろう。このときの目標値は、ユーザーが任意に設定できるものとするが、数値での設定はわかりづらいので、原稿サイズと画質（高中低等）を加味した直感的に分かりやすいメニューから選択することで、決定するようにすれば良いであろう。20

#### 【 0 1 6 0 】

また、符号化部の符号化パラメータとして量子化ステップを例にして説明したが、圧縮率の異なるデータが混在した際に、それらの間での画質が違和感が発生しないようにする限りは、他のパラメータを用いても良い。但し、例えば、図 1 の構成においては、再符号化部 109 からの再度符号化するデータが、パラメータ変更後の符号化部 102 からの符号化データと実質的に同じにするには、上記実施形態に示す如く、量子化ステップを増加する手法が好ましい。30

#### 【 0 1 6 1 】

また、上記の通り、本発明は、汎用装置上で動作するアプリケーションプログラムによって実現できるものであるので、本発明はコンピュータプログラムをも含むものである。また、コンピュータプログラムは、通常、フロッピーディスクや C D R O M 等の記憶媒体を装置にセットしてコピー或いはインストールことで行われるので、かかる記憶媒体も本発明の範疇に当然に含まれる。

#### 【 0 1 6 2 】

また、実施形態では、スキャナから画像データを入力するものとして説明したが、ホストコンピュータ上で動作するプリンタドライバに適用しても良い。プリンタドライバに適用する場合には、上位処理（アプリケーション等）から印刷対象のデータを受信したときに、その時点で、そのデータが中間調画像か、文字・線画かは勿論は判別できるので、像域情報生成処理にかかる構成を省くか、或いはより簡素なものとすることができます。40

#### 【 0 1 6 3 】

また、本発明は、コンピュータプログラムと適当なハードウェア（符号化回路等）の組み合わせにも適用できる。

#### 【 0 1 6 4 】

以上説明したように本第 1 、第 2 の実施形態によれば、可逆符号化した像域情報を復号す50

る復号化手段と、像域情報を該像域情報の情報エントロピーが少なくなるように該像域情報の一部を書き換え、或いは削除する像域情報変換手段と、前記復号化手段で復号した像域情報を前記像域情報変換手段で変換した該像域情報を再び可逆符号化する再符号化手段と、前記情報変換手段を備えた可逆符号化手段と、少なくとも1ページ分の画像データに付随する像域情報を符号化したデータを格納することが可能な格納手段とを有し、前記符号化データの量に応じて、前記可逆変換手段が備えた像域情報変換手段と、再符号化手段の前段にある像域情報変換手段の双方を制御することにより、1ページ分の像域情報を所望の符号化データ量に収めることができる。

#### 【0165】

<第3の実施形態>

10

本第3の実施形態を説明する。本第3の実施形態は、先に説明した第1の実施形態とほぼ同様の動作を行うが、その構造が異なる。図22はそのブロック構成図である。図1と同符号の構成要素については、その動作（画像データの符号化処理）が同じであるので、その説明は省略し、以下、本第3の実施形態における像域情報の符号化部分について説明する。

#### 【0166】

イメージスキャナやページ記述言語レンダリング等から入力部101を介して入力した画像データ（多値カラー画像データ）は、符号化部102に供給されると共に像域情報生成部2201にも供給される。尚、上記の如く、符号化部102以降は図1と同様であるので、その説明は省略する。

20

#### 【0167】

像域情報生成部2201では、入力した画像データから像域情報を生成する。入力対象がスキャナの場合には、先に説明した第1の実施形態における像域情報生成部1701と同様の処理を行えば良いし、PDLレンダリングからの場合にはPDLレンダリングでイメージ展開する際に各画素毎に像域情報が判明しているので、その情報を活用すればよいであろう。

#### 【0168】

生成された像域情報はブロック化ユニット2202において、まとめて符号化するデータサイズ、例えばM×Nのサイズへブロック化する。

30

画像データの圧縮を利用するJPEGのような多値の非可逆圧縮は、2値データの集まりである像域情報の圧縮に用いることに不都合があるので、可逆圧縮であるJBIG或いはPacK Bits等のランレングス符号化を用いて、可逆符号化部2203にて可逆符号を行う。

#### 【0170】

可逆符号した像域情報は、メモリ制御部2204を経由して、第1のメモリ104に格納される。

#### 【0171】

図27は、符号化の初期状態におけるメモリの状態を示している。図4と異なるのは、図示の如く、像域情報の記憶エリアが設けられている点である。符号化された像域情報は、メモリ制御部2204を経由して、この記憶エリアに順次、格納されていく。

40

#### 【0172】

先に説明をした通り、符号化シーケンス制御部108では、第1のカウンタ107のカウント値がある設定値に達したかどうかを検出し、その設定値に達したことを検出すると所定の制御信号を出力し、符号化部102に対して今までより、高い圧縮率で画像データに対して、符号化を行なうよう制御する。

#### 【0173】

そして、本第3の実施形態では、それと同時に符号化シーケンス制御部108は、上述の画像データの再符号化のタイミングに同期して、像域情報の再符号化の指示を符号化制御部2205に指示する。

50

**【0174】**

符号化制御部2205は、第1のメモリ104から符号化した像域データを読み出し、該データを可逆符号再符号化部2206に送るよう、メモリ制御部2204へ制御信号を出力する。

**【0175】**

可逆符号再符号化部2206は、符号化データを受け取るとそれを復号し、複数の属性フラグデータの一部を廃棄するか固定値に置き換えた後、再び可逆符号化を行なう。属性フラグの一部を固定値に置き換えた場合でも情報エントロピーが低下するため、ランレンジス符号化後のデータ量は減少する。再符号化後の属性データは、第1のメモリ104に再び格納する。

10

**【0176】**

一方、符号化制御部2205は、可逆符号再符号化部2206において情報量を減らした属性フラグデータと、同じ情報の属性フラグデータを符号化するよう、可逆符号化部2203が備えた像域情報変換処理部に対して属性フラグの一部を廃棄、或いは固定値に置き換えるよう制御信号を送り、符号化処理を継続させる。

**【0177】**

前記再符号化する像域フラグデータが無くなり再符号化処理が終了するまで続ける。

**【0178】**

符号化シーケンス制御部108が、カウンタ107のカウント値がある設定値に達したかどうかを検出し、目標値をオーバーするたびに、画像データの再符号化を行うたびに、それと同期して像域フラグの廃棄する属性フラグを増やすことで、像域データの符号量を段階的に減らすことができ、該像域データの符号量を目標値以内に收めることができとなる。

20

**【0179】**

次に、本第3の実施形態の処理内容（像域情報の符号化処理）を図23のフローチャートに従って説明する。本第3の実施形態の処理は、上述したように、大きく2つの処理に分かれる。1つは可逆符号化処理、もう1つは再符号化処理である。

**【0180】**

符号化処理は、ステップS2305の像域情報変換処理とステップS2307の可逆符号化処理で実現する。また、再符号化処理は、ステップS2315の復号化処理と、ステップS2317の像域情報変換処理、それにステップS2319の再符号化処理で実現する。

30

**【0181】**

その他のステップは、S2313の像域情報変更処理の変更を除くと、条件分岐と開始及び終了ステップである。

**【0182】**

まず、ステップS2301から、像域情報の符号化処理を開始する。次のステップS2303で、画像1ページに対する像域情報の入力が全て終了しているかどうかを検知し、終了していればステップS2321へ行き、終了していない場合、すなわち、像域情報の入力がある場合には、ステップS2305の像域情報変換処理を行なう。

40

**【0183】**

再符号化処理がまだ一度も起動していない初期状態では、像域変換処理では何も処理せず、入力された像域情報をそのまま次のステップS2307の可逆符号化処理で符号化する。

**【0184】**

次のステップS2309では、再符号化処理中であるかどうかを判定し、再符号化処理中であれば、ステップS2315～S2319から成る再符号化処理を行なうが、再符号化処理中でなければ、ステップS2311へ進み、画像データの再符号化が行われるのかどうかを判定する。

**【0185】**

50

符号化シーケンス制御部 108 から所定の目標値に対して符号化データが、オーバーしたという制御信号を受けていなければ、ステップ S2303 に戻り、符号化処理を繰り返し、オーバーしたという制御信号を受ければ、像域情報変換処理における処理内容を変更する。該変更により、ステップ S2305、S2317 における像域変換処理の内容が変わる。具体的には、初期状態では、入力された像域情報を 1 ビットも無効にすることなく全て符号化してきた。すなわち、像域変換処理では何もしていなかったが、処理内容を変更後は、少なくとも像域情報中の 1 ビットを廃棄するか固定値に置き換えると言った処理を、前記像域変換処理で行なう。例えば、文字線画を示すビットを全て中間調であることを示すビットに変換する。この後、処理内容を変更するごとに、廃棄または固定値に置き換える像域フラグのビット数を増やしていく（ラン長が長くなり易い状態にする）。 10

#### 【0186】

該像域情報変換処理内容の変更後は、再符号化処理を行なう。再符号化処理では、ステップ S2315 にて、符号化した像域情報を復号化処理して符号化前のデータに戻す。次のステップ S2317 にて、前記像域情報変換処理を行ない、像域情報の一部を廃棄または固定値に置き換える。そして、次のステップ S2319 にて、変換した像域情報を再び可逆符号化する。

#### 【0187】

上記再符号化処理が終わったら、ステップ S2303 に戻り、像域情報の入力があれば、それを符号化処理する。画像データの符号量が一度でも設定値をオーバーし再符号化が開始されると、それと連動して、像域情報変換処理内容が変更され、ステップ S2305 の像域情報変換処理にて、一部の像域フラグデータが廃棄または固定値に置き換えられた後に、ステップ S2307 にて、可逆符号化される。 20

#### 【0188】

像域情報の入力が全て終了しても、再符号化処理が継続中であれば、ステップ S2303 からステップ S2321 へ移り、そこで、再符号化処理中であることを判定して、前記ステップ S2315～S2319 から成る再符号化処理を行なう。

#### 【0189】

像域情報の入力が全て終了し、且つ、再符号化処理もしていなければ、ステップ S2323 へ移り、像域情報の符号化処理を終了する。

#### 【0190】

また、再度、画像データの符号量が一度でも設定値をオーバーした場合には、今度は、像域情報中の有彩色／無彩色を示すビットを全て有彩色に変更する。 30

#### 【0191】

次に、1 画素あたり 2 ビットの像域情報を 6 ビットデータ “000000” を付加して 8 ビット化した後、可逆符号として packbit s 符号化を行なった場合の、本第 3 の実施形態における具体的な処理内容を、図 24 を用いてさらに詳しく説明する。

#### 【0192】

Packbits 符号化する前の 8 ビットデータは、図 24 (a) に示すように上位 6 ビットは全て 0 で、下位 2 ビットの上位側に、対応する画素データが文字線画領域か、中間調かフラグビット、下位側には有彩色か無彩色を表わすフラグビットが割り当てられている。よって、該 8 ビットデータが取り得る値は、0 以上 3 以下の値である。 40

#### 【0193】

像域情報生成部 2201 からは、上記 8 ビットのデータが画素単位で出力されるものとする。具体的な出力データとして、図 24 (b) に示すデータを考える。

#### 【0194】

これを packbits 符号化すると同図 (c) に示すデータに圧縮される。圧縮後のデータで負の値は連続するデータの個数を表わし、非連続データの個数は正の値で表わしている。これらは長さ情報と言うもので、該長さ情報のサインビットから連続データが続くのか、非連続データが続くのかを判別することが出来るようになっている。圧縮後の各データは同図 (b) と同じく 8 ビット (1 バイト) である。1 バイトの長さ情報で、表わす 50

ことができる最大値は 255 の半分の約 128 であり、長さ情報がそれ以下の場合は 1 組の長さ情報とそれに続く像域フラグデータ群で符号化でき、それを超える場合は複数の組の、長さ情報 + 像域フラグデータ群、に分けて符号化される。

#### 【0195】

図 24 (c) の圧縮データを詳しく見てみることにする。最初の長さ情報 “ -4 ” はマイナスの値なので、上述したように連続データの連続個数を表わし、長さ情報直後の像域フラグデータ “ 1 ” が 4 つ続くことを表わしている。

#### 【0196】

次のデータ “ 4 ” はまた長さ情報であるが、今度はプラスの値なので非連続データが 4 つ続くことを示している。よって、前記 “ 4 ” に続く 4 つのデータ “ 2 , 3 , 2 , 3 ” が非連続データを表わす。同図 (c) では、長さ情報と像域フラグデータとが区別し易いように、プラスの長さ情報のみ下線を引いている。10

#### 【0197】

上記非連続データの次の “ -5 ” は又、連続データの長さ情報で、該長さ情報直後の像域フラグデータ “ 2 ” が 5 つ続くことを表わしている。次の下線付きのデータ “ 3 ” は非連続データの長さ情報で、後続する 3 つのデータ “ 1 , 0 , 1 ” が像域フラグデータであり、さらに次の “ -6 , 0 ” は、データ “ 0 ” が 6 個連続することを示している。

#### 【0198】

上記圧縮データを可逆符号再符号化部 1715 で再符号化処理するとどのようになるのかを、図 24 (d) (e) を用いて説明する。ここでは、再符号化処理において、有彩色・無彩色フラグを “ 1 ” に固定してすべて有彩色にしてしまう例を説明する。20

#### 【0199】

符号化された像域データは一旦復号され、同図 (b) のデータに戻された後、上記フラグデータの置き換えが行なわれ、同図 (d) のデータに変換される。そして、変換されたデータを再び P a c k b i t s 符号化することによって、同図 (e) の符号化データが得られる。再符号化前の 15 バイトの符号化データが再符号化後には 6 バイトに減少することが解かる。

#### 【0200】

上記の再符号化処理を行なった後にも、画像データの符号化において、全符号量の計数値が設定された目標値を再びオーバーし、画像データの再符号化を行う場合には、符号化シーケンス制御部 108 から像域データも再符号化する旨の制御信号を受け、前記再符号化処理が終了していれば、直ちに次の新たな再符号化処理を開始する。前記再符号化処理が終了していなければ、該再符号化処理が終了後、直ちに次の新たな再符号化処理を開始する。30

#### 【0201】

新たな再符号化処理では、残りの 1 ビットの像域フラグも “ 1 ” に置き換える。これにより、すべての像域フラグデータ (8 ビット) の値は “ 3 ” となり、データのバイト数を N とすると、符号化後のデータ量はおよそ (2N / 128) + 2 バイトとなる。

#### 【0202】

これは、連続データの個数が 128 個を超えるたびに、あらたな 2 バイト 1 組の符号化データ (長さ情報と連続データ) が増えるためである。40

#### 【0203】

P a c k b i t s の符号化回路や復号回路それにデータ変換回路はそれぞれ公知の技術であるため、僻別の拘路構成についての説明は省略する。

#### 【0204】

上記説明では簡単化のため、各画素の像域フラグを 2 ビットとして説明したが、前述したように像域フラグとして他の情報もいくつかある。上記再符号化処理では、2 ビットの像域フラグデータでは最大 2 回の再符号化、4 ビットの像域フラグデータでは最大 4 回の再符号化処理が可能であり、像域フラグのビット数が多い程、再符号化処理の回数を増やすことができ、符号量を多段階で制御することができる。50

**【0205】**

上述したように像域フラグデータの可逆符号化処理は、画像データの圧縮符号化処理とは独立に制御され、それぞれに目標符号量以内のデータに収められる。

**【0206】**

符号化された2種類のデータは、外部に接続されるネットワーク機器、画像出力装置や大容量記憶装置等へ出力する際に多重化する。該多重化を考慮して、前記2種類のデータを符号化処理する単位を同じサイズに合わせておき、1単位を符号化して生成される符号化データを1つのパケットあるいはファイルとして管理・格納する。多重化する際に、画像位置が同じ2種類のパケットデータを、例えば画像データ・像域データの順に連結して、1パケット化し、外部へ出力する。

10

**【0207】**

前記2つの符号化処理は独立に制御されるので、画像データの圧縮符号化処理部が他の構成であっても構わない。よって、像域フラグデータを可逆符号化処理するユニット2201～2206を前記図2の構成に対して付加しても、同じように処理することが可能である。その構成を図25に示しておくが、ユニット2201～2206の動作は図22とまったく同じであり、画像データ圧縮部分の動作も図2の動作とまったく同じである。

**【0208】**

<第4の実施形態>

第4の実施形態では、像域情報のビット数が第3の実施形態と同じでも、再符号化処理の回数を増やせて、符号量を細かく制御することが出来る処理方法について説明する。本実施形態の構成も前記第1の実施形態と同じであり、違うのは、可逆符号化部1703と可逆符号再符号化部1706における像域フラグデータの縮退方法である。

20

**【0209】**

前記第1の実施形態では、像域フラグデータを1ビットずつ固定値に置き換える、すなわち、データを1ビットずつ縮退させるものであったが、本実施形態では、状態数を縮退させる。例えば、2ビットの像域フラグでは、4状態を表わすことができるが、これを1回目の再符号化処理で3状態に縮退させ、2回目の再符号化処理で2状態に縮退することで、符号化前の情報エントロピーを減らし、符号化後のデータ量（符号量）を細かく減らしていくものである。

**【0210】**

30

状態数という言葉を用いて、先の第3の実施形態を表現すると、像域フラグデータを再符号化するごとに状態数を半分に減らす、と言うことが出来る。

**【0211】**

1回の再符号化処理で、第3の実施形態が状態数を半分に減らすのに対し、本実施形態は状態数を1つずつ減らすわけであるから、符号量を細かく減らせるのは当然である。

**【0212】**

本実施形態の処理結果を図26（b）（c）（d）（e）に示し、それについて説明する。

**【0213】**

図26（a）は、図24（b）に示した像域フラグデータと同じ、2ビットの4状態の全てが存在するデータである。該4状態をあらためて列挙すると、以下のようになる。

40

**【0214】**

- (1) 有彩色の文字部（データ“3”に対応）
- (2) 無彩色の文字部（データ“2”に対応）
- (3) 有彩色の非文字部（データ“1”に対応、有彩色の画像部とも言う）
- (4) 無彩色の非文字部（データ“0”に対応、無彩色の画像部とも言う）

本実施形態では、1回目の再符号化処理で、上記4状態の内、(3)有彩色の非文字部と(4)無彩色の非文字部、の2状態を合体して1つの状態(3')非文字部に縮退させる。

これにより、以下の3つの状態となる。

50

**【0215】**

- (1) 有彩色の文字部
- (2) 無彩色の文字部、
- (3') 非文字部

具体的には、データ“0”を“1”に置き換えることで、上記状態の縮退を行なう。状態縮退後のデータは図26(b)に示すデータに変わる。これをp a c k b i t s符号化すると図26(c)に示す符号化データになる。再符号化前の符号化データ図24(c)よりも多少符号量が少なくなっていることが解かる。

**【0216】**

2回目の再符号化処理では、(1) 有彩色の文字部と(2) 無彩色の文字部、の2状態を合体して1つの状態(1') 文字部に縮退させる。これにより、以下の2状態となる。

**【0217】**

- (1') 文字部
- (3') 非文字部

今度は、データ“2”を“3”に置き換えることで、上記状態の縮退を行なう。状態縮退後のデータは図26(d)に示すデータに変わる。このデータは図24(d)のデータと同じである。これをP a c k b i t s符号化した符号化データ図26(e)は当然図24(e)と同じ符号化データになる。

**【0218】**

第3の実施形態では、再符号化前の15バイトのデータが1回の再符号化で6バイトまで減少したが、本第4の施形態では、1回目の再符号化で11バイトに減少し、2回目の再符号化でやっと6バイトに減少する。結局、第4の実施形態では、第3の実施形態では得られない細かく変化して減少する符号量を得ることが出来るので、目標符号量に近い像域フラグの圧縮データを得ることができる。

**【0219】**

以上説明したように第3、第4の実施形態によれば、先に説明した第1、第2の実施形態と同様、1ページ分の像域情報を所望の符号化データ量に収めることができる。

**【0220】**

## &lt;第5の実施形態&gt;

本第5の実施形態を説明する。本第5の実施形態は、先に説明した第1の実施形態とほぼ同様の動作を行うが、その構造が異なる。図28はそのブロック構成図である。図1と同符号の構成要素については、その動作(画像データの符号化処理)が同じであるので、その説明は省略し、以下、本第5の実施形態における像域情報の符号化部分について説明する。

**【0221】**

イメージスキヤナやページ記述言語レンダリング等から入力部101を介して入力した画像データ(多値カラー画像データ)は、符号化部102に供給されると共に像域情報生成部2801にも供給される。尚、上記の如く、符号化部102以降は図1と同様であるので、その説明は省略する。

**【0222】**

像域情報生成部2801では、入力した画像データから像域情報を生成する。入力対象がスキヤナの場合には、先に説明した第1の実施形態における像域情報生成部1701と同様の処理を行えば良いし、PDLレンダリングからの場合にはPDLレンダリングでイメージ展開する際に各画素毎に像域情報が判明しているので、その情報を活用すればよいであろう。

**【0223】**

生成された像域情報はブロック化ユニット2803において、まとめて符号化するデータサイズ、例えばM×Nのサイズへブロック化する。

**【0224】**

画像データの圧縮を利用するJPEGのような多値の非可逆圧縮は、2値データの集まり

10

20

30

40

50

である像域情報の圧縮に使うのは不都合であり、可逆圧縮である JBIG、或いは Pack Bits 等のランレンジング符号化を用い、可逆符号化部 2805 にて像域情報を可逆符号化する。

#### 【0225】

符号化した像域情報は、第3のメモリ制御部 2807 を経由して、第1のメモリ 104 に格納される。また、これと同時に、可逆符号化部から出力される符号データの量は第4のカウンタ 2811 にて累積カウントされ、該カウント値は符号化制御部 2813 に送られる。

#### 【0226】

図33は、符号化の初期状態でのメモリの状態を示している。図4と異なるのは、像域情報の記憶エリアが設けられている点である。基本的には、符号化後の像域情報は、この記憶エリアに順次保持していくが、後述するように、符号化後の画像データをこの記憶エリアに記憶させることも可能である。また、符号化後の像域データは、この記憶エリアに収まらない場合でも、画像データの記憶エリアの方に余剰がある場合には、その余剰分の記憶エリアに保持することも、もちろん可能である。

#### 【0227】

符号化制御部 2813 と、符号化シーケンス制御部 108 は、それぞれ第4のカウンタ 2811 と、第1のカウンタ 107 からのカウント値を、第6のカウンタ 2809 へ送る。

#### 【0228】

第6のカウンタは、画像データの符号化、像域情報の符号化のカウントを合計し、そのカウント値を、画像・像域符号化制御部 2819 へ送る。

#### 【0229】

画像・像域符号化制御部 2819 内のレジスタには、画像データおよび像域データの合計目標値が設定されており、前記画像データ、像域データの符号量の合計カウント値が、目標値をオーバーしたとき、符号化制御部 2813、符号化シーケンス制御部 108 に対して、再符号化処理を行うよう制御信号を出力する。

#### 【0230】

符号化制御部 2813 は、第1のメモリ 104 から符号化した像域データを読み出し、像域データを可逆符号再符号化部 2815 に送るよう、メモリ制御部 2807 へ制御信号を出力する。

#### 【0231】

符号化シーケンス制御部 108 は、第1のメモリ 104 に格納済みのデータを廃棄するよう第1のメモリ制御部 103 に制御信号を出力する。上記第1のメモリ制御部 103 は、該制御信号に基づいて、メモリアドレスカウンタをクリアするか、あるいは符号化データ管理テーブルをクリアすることにより、前記格納データを廃棄する。そして符号化シーケンス制御部 108 は、符号化部 102 に対して今までより、高い圧縮率で符号化を行なうよう制御し、再符号化処理を行う。

#### 【0232】

すなわち、第1のカウンタ 107 のカウント値が、符号化シーケンス制御部 108 内のレジスタ内に記憶されている画像データの符号量の目標値をオーバーしても、あるいは、第4のカウンタ 2811 のカウント値が、符号化制御部 2813 内のレジスタ内に記憶されている像域データの符号量の目標値をオーバーしても、画像データの符号量の目標値と像域データの符号量の目標値の合計に対して、画像データと像域データの符号量の合計がオーバーしていなければ、符号化処理を継続させる。

#### 【0233】

これによって、例えば、画像データは若干、画像データの目標値をオーバーしてしまったが、それに対応する像域情報の符号量が小さいため、第1のメモリ 104 内の像域データに割り当てられている記憶エリアを画像データに割り当てることで、画像データと像域情報の符号量の合計が、画像データと像域データの符号量の目標値内に納まるのであれば、画像データを再符号化する必要がなくなる。さらなる高圧縮を行う必要がなくなり、画質向

10

20

30

40

50

上、パフォーマンス向上につながる。

**【0234】**

画像データの再符号化処理は、画像・像域符号化制御部2819から符号化シーケンス制御部108に対して再符号化処理の制御信号が出力されると、符号化シーケンス制御部108は、図1の説明で先に述べた通りの再符号化処理を行うための制御を行う。

**【0235】**

像域情報の再符号化処理は、画像・像域符号化制御部2819から符号化制御部2813に対して再符号化処理の制御信号が出力されると、符号化制御部2813は、第1のメモリ104から符号化した像域データを読み出し、該データを可逆符号再符号化部2815に送るよう、メモリ制御部2807へ制御信号を出力する。

10

**【0236】**

可逆符号再符号化部2815は、符号化データを受け取るとそれを復号化し、複数の属性情報の一部を廃棄するか固定値に置き換えた後、再び可逆符号化を行なう。属性フラグの一部を固定値に置き換えた場合でも情報エントロピーが低下するため、ランレンジス符号化後のデータ量は減少する。再符号化後の属性データは、第1のメモリ104に再び格納すると共に、その符号量を第5のカウンタ2817にてカウントする。

**【0237】**

一方、符号化制御部2813は、可逆符号再符号部2815と同様、可逆符号化部2805に対して、それ以降に入力される属性情報の一部を廃棄、或いは固定値に置き換えるよう制御信号を送り、符号化処理を継続させる。それと同時に、第4のカウンタ2811にも制御信号を送り、それまでカウントして保持していた値をリセットさせ、属性情報の一部を廃棄、或いは固定値に置き換えた後に符号化処理して生成する符号量を新たにカウントさせる。

20

**【0238】**

前記再符号化する像域情報が無くなり再符号化処理が終了したら、第5のカウンタ2817の計数値を第4のカウンタ2811に転送して加算する。それによって、カウンタ2811には、属性情報を一部廃棄、或いは固定値に置き換えた後の可逆符号化データの全符号量が計数される。

**【0239】**

画像データ、像域情報の再符号化を行うたびに、各々の符号量のカウントを行い、画像データ、像域情報の符号量の合計値が、画像・像域符号化制御部2819内のレジスタに設定された画像データ、像域情報の目標値の合計をオーバーするたびに、画像データの再符号化処理、および、像域情報の再符号化処理を行う。像域情報の再符号化は、廃棄する属性フラグを増やすことで、像域データの符号量を投階的に減らすことができ、該像域情報の符号量を目標値以内に収めることができる。

30

**【0240】**

次に、本実施形態の処理内容を図29のフローチャートに従って説明する。本実施形態の処理は、上述したように、大きく2つの処理に分かれる。1つは可逆符号化処理、もう1つは再符号化処理である。

**【0241】**

符号化処理は、ステップS2905の像域情報変換処理とステップS2907の可逆符号化処理で構成される。また、再符号化処理は、ステップS2915の復号化処理と、ステップS2917の像域情報変換処理、それにステップS2919の再符号化処理で構成される。その他のステップは、S2913の像域情報変更処理の変更を除くと、条件分岐と開始及び終了ステップである。

40

**【0242】**

まず、ステップ2901から、像域情報の符号化処理を開始する。次のステップ2903で、画像1ページ(1つの画像)に対する像域情報の入力が全て終了しているかどうかを検知し、終了していればステップS2921へ行き、終了していない場合、すなわち、像域情報の入力がある場合には、ステップS2905の像域情報変換処理を行なう。

50

**【0243】**

再符号化処理がまだ一度も起動していない初期状態では、像域変換処理では何も処理せず、入力された像域情報をそのまま次のステップS2907の可逆符号化処理で符号化する。

**【0244】**

次のステップS2909では、再符号化処理中であるかどうかを判定し、再符号化処理中であれば、前記ステップS2915～S2919から成る再符号化処理を行なうが、再符号化処理中でなければ、ステップS2911へ行き、画像・像域符号化制御部において符号化した画像データ、像域情報の合計データ量（合計符号量）が設定値をオーバーしたかどうかを判定する。

10

**【0245】**

オーバーしていなければ、ステップS2903に戻り、符号化処理を繰り返し、オーバーしていれば、像域情報変換処理における処理内容を変更する。該変更により、ステップS2905、S2917における像域変換処理の内容が変わる。具体的には、初期状態では、入力された像域フラグデータを1ビットも無効にすることなく全て符号化してきた。すなわち、像域変換処理では何もしていなかったが、処理内容を変更後は、少なくとも像域情報中の1ビットのフラグデータを廃棄するか固定値に置き換えるといった処理を、前記像域変換処理で行なう。この後、処理内容を変更するごとに、廃棄または固定値に置き換える像域フラグのビット数を増やしていく。

**【0246】**

20

該像域情報変換処理内容の変更後は、再符号化処理を行なう。再符号化処理では、ステップS2915にて、既に符号化した像域情報を復号化処理して符号化前のデータに戻す。そして、次のステップS2917にて、像域情報変換処理を行ない、像域情報の一部を廃棄または固定値に置き換える。そして、次のステップS2919にて、変換した像域情報を再び可逆符号化する。

**【0247】**

上記再符号化処理が終わったら、ステップS2903に戻り、像域情報の入力があれば、それを符号化処理する。画像・像域符号化制御部において、画像データ、像域データの符号量の合計が一度でも設定値をオーバーすると、像域情報変換処理内容が変更され、ステップS2905の像域情報変換処理にて、一部の像域フラグデータが廃棄または固定値に置き換えられた後に、ステップS2907にて、可逆符号化される。

30

**【0248】**

像域情報の入力が全て終了しても、再符号化処理が継続中であれば、ステップS2903からステップS2921へ移り、そこで、再符号化処理中であることを判定して、前記ステップS2915、S2919から成る再符号化処理を行なう。

**【0249】**

像域情報の入力が全て終了し、且つ、再符号化処理もしていなければ、ステップS2923へ移り、像域情報の符号化処理を終了する。

**【0250】**

40

次に、1画素あたり2ビットの像域フラグデータに6ビットデータ“000000”を付加して8ビット化した後、可逆符号としてP a c k b i t s 符号化を行なった場合の、本実施形態における具体的な処理内容を、図30を用いてさらに詳しく説明する。

**【0251】**

P a c k b i t s 符号化する前の8ビットデータは、図30(a)に示すように上位6ビットは全て0で、下位2ビットの上位側に、対応する画素データが文字部かそうでないかを表わすフラグ、下位側には有彩色か無彩色を表わすフラグデータが入っている。よって、該8ビットデータが取り得る値は、0以上3以下の値である。

**【0252】**

像域情報生成部2801からは、上記8ビットのデータが画素単位で出力されるものとする。具体的な出力データとして、図30(b)に示すデータを考える。

50

**【0253】**

これを P a c k b i t s 符号化すると同図 (c) に示すデータに圧縮される。圧縮後のデータで負の値は連続するデータの個数を表わし、非連続データの個数は正の値で表わしている。これらは長さ情報と言うもので、該長さ情報のサインビット (M S B) から連続データが続くのか、非連続データが続くのかを判別することが出来るようになっている。圧縮後の各データは同図 (b) と同じく 8 ビット (1 バイト) である。1 バイトの長さ情報で、表わすことができる最大値は 255 の半分の約 128 であり、長さ情報がそれ以下の場合は 1 組の長さ情報とそれに続く像域情報のフラグデータ群で符号化でき、それを超える場合は複数の組の、長さ情報 + 像域フラグデータ群、に分けて符号化される。

**【0254】**

10

図 30 (c) の圧縮データを詳しく見てみることにする。最初の長さ情報 “-4” はマイナスの値なので、上述したように連続データの連続個数を表わし、長さ情報直後の像域フラグデータ “1” が 4 つ続くことを表わしている。

**【0255】**

次のデータ “4” はまた長さ情報であるが、今度はプラスの値なので非連続データが 4 つ続くことを示している。よって、前記 “4” に続く 4 つのデータ “2, 3, 2, 3” が非連続データを表わす。同図 (c) では、長さ情報と像域フラグデータとが区別し易いように、プラスの長さ情報のみ下線を引いている。

**【0256】**

20

上記非連続データの次の “-5” は又、連続データの長さ情報で、該長さ情報直後の像域フラグデータ “2” が 5 つ続くことを表わしている。次の下線付きのデータ “3” は非連続データの長さ情報で、後続する 3 つのデータ “1, 0, 1” が像域フラグデータであり、さらに次の “-6, 0” は、データ “0” が 6 個連続することを示している。

**【0257】**

上記圧縮データを可逆符号再符号化部 2815 で再符号化処理するとどのようになるのかを、同図 (d)、(e) を用いて説明する。ここでは、再符号化処理において、有彩色・無彩色フラグを “1” に固定してすべて有彩色にする例を示す。

**【0258】**

30

符号化された像域データは一旦復号され、図 30 (b) のデータに戻された後、上記フラグデータの置き換えが行なわれ、同図 (d) のデータに変換される。そして、変換されたデータを再び P a c k b i t s 符号化することによって、同図 (e) の符号化データが得られる。再符号化前の 15 バイトの符号化データが再符号化後には 6 バイトに減少することが解かる。

**【0259】**

上記の再符号化処理を行なったにもかかわらず、画像データ、像域データの符号量の合計である全符号量の計数値が画像・像域符号化制御部 2819 内のレジスタに設定された画像データと像域データの符号量の合計目標値を再びオーバーした場合には、前記再符号化処理が終了していれば、直ちに次の新たな再符号化処理を開始する。前記再符号化処理が終了していなければ、該再符号化処理が終了後、直ちに次の新たな再符号化処理を開始する。

40

**【0260】**

新たな再符号化処理では、残りの 1 ビットの像域フラグも “1” に置き換える。これにより、すべての像域フラグデータ (8 ビット) の値は “3” となり、データのバイト数を N とすると、符号化後のデータ量はおよそ (2N / 128) + 2 バイトとなる。

**【0261】**

これは、連続データの個数が 128 個を超えるたびに、あらたな 2 バイト 1 組の符号化データ (長さ情報と連続データ) が増えるためである。

**【0262】**

P a c k b i t s の符号化回路や復号回路それにデータ変換回路はそれぞれ公知の技術であるため、個別の回路構成についての説明は省略する。

50

**【0263】**

上記説明では簡単化のため、各画素の像域情報の有意なビット数を2ビットとして説明したが、前述したように像域フラグとして他の情報もいくつかある。上記再符号化処理では、2ビットの像域フラグデータでは最大2回の再符号化、4ビットの像域フラグデータでは最大4回の再符号化処理が可能であり、像域フラグのビット数が多い程、再符号化処理の回数を増やすことができ、符号量を多段階で制御することができる。

**【0264】**

上述したように像域フラグデータの可逆符号化処理は、画像データの圧縮符号化処理とは独立に制御され、それぞれに目標符号量以内のデータに収められる。

**【0265】**

符号化された2種類のデータは、外部に接続されるネットワーク機器、画像出力装置や大容量記憶装置等へ出力する際に多重化する。該多重化を考慮して、前記2種類のデータを符号化処理する単位を同じサイズに合わせておき、1単位を符号化して生成される符号化データを1つのパケットあるいはファイルとして管理・格納する。多重化する際に、画像位置が同じ2種類のパケットデータを、例えば画像データ・像域データの順に連結して、1パケット化し、外部へ出力する。

**【0266】**

前記2つの符号化処理は独立に制御されるので、画像データの圧縮符号化処理部が他の構成であっても構わない。よって、像域フラグデータを可逆符号化処理するユニット2801、2819を前記図2の構成に対して付加しても、同じように処理することが可能である。その構成を図31に示しておくが、ユニット2801～2819の動作は図28とまったく同じであり、画像データ圧縮部分の動作も図2の動作と同じである。

**【0267】**

<第6の実施形態>

第2の実施形態では、像域情報のビット数が前記第1の実施形態と同じでも、再符号化処理の回数を増やせて、符号量を細かく制御することが出来る処理方法について説明する。本実施形態の構成も前記第5の実施形態と同じであり、異なるのは、可逆符号化部2805と可逆符号再符号化部2815における像域フラグデータの縮退方法である。

**【0268】**

前記第1の実施形態では、像域情報を1ビットずつ固定値に置き換える、すなわち、データを1ビットずつ縮退させるものであったが、本実施形態では、状態数を縮退させる。例えば、2ビットの像域フラグでは、4状態を表わすことができるが、これを1回目の再符号化処理で3状態に縮退させ、2回目の再符号化処理で2状態に縮退することで、符号化前の情報エントロピーを減らし、符号化後のデータ量（符号量）を細かく減らしていくものである。

**【0269】**

状態数という言葉を用いて、前記第5の実施形態を表現すると、像域情報を再符号化するごとに状態数を半分に減らす、と言うことが出来る。

**【0270】**

1回の再符号化処理で、第5の実施形態が状態数を半分に減らすのに対し、本第6の実施形態は状態数を1つずつ減らすわけであるから、符号量を細かく減らせるのは当然である。

**【0271】**

本実施形態の処理結果を図32(b)(c)(d)(e)に示し、それについて説明する。

**【0272】**

同図(a)は、図30(b)に示した像域フラグデータと同じ、2ビットの4状態の全てが存在するデータである。該4状態をあらためて列挙すると、以下のようになる。

**【0273】**

(1) 有彩色の文字線画部（データ“3”に対応）

10

20

20

30

40

50

(2) 無彩色の文字線画部(データ“2”に対応)

(3) 有彩色の非文字線画部(中間調部)(データ“1”に対応、有彩色の画像部とも言う)

(4) 無彩色の非文字線画部(中間調部)(データ“0”に対応、無彩色の画像部とも言う)

本実施形態では、1回目の再符号化処理で、上記4状態の内、(3)有彩色の非文字部と(4)無彩色の非文字部、の2状態を合体して1つの状態(3')非文字部に縮退させる。

#### 【0274】

これにより、以下の3つの状態となる。

10

#### 【0275】

(1) 有彩色の文字部

(2) 無彩色の文字部、

(3') 非文字部

具体的には、データ“0”を“1”に置き換えることで、上記状態の縮退を行なう。状態縮退後のデータは図32(b)に示すデータに変わる。これをP a c k b i t s符号化すると同図(c)に示す符号化データになる。再符号化前の符号化データは図30(c)よりも多少符号量が少なくなっていることが解かる。

#### 【0276】

2回目の再符号化処理では、(1)有彩色の文字部と(2)無彩色の文字部、の2状態を合体して1つの状態(1')文字部に縮退させる。これにより、以下の2状態となる。

20

#### 【0277】

(1') 文字部

(3') 非文字部

今度は、データ“2”を“3”に置き換えることで、上記状態の縮退を行なう。状態縮退後のデータは図32(d)に示すデータに変わる。このデータは図30(d)のデータと同じである。これをP a c k b i t s符号化した符号化データである図32(e)は当然図30(e)と同じ符号化データになる。

#### 【0278】

第5の実施形態では、再符号化前の15バイトのデータが1回の再符号化で6バイトまで減少したが、本第6の実施形態では、1回目の再符号化で11バイトに減少し、2回目の再符号化でやっと6バイトに減少する。結局、第6の実施形態では、第5の実施形態では得られない細かく変化して減少する符号量を得ることが出来るので、目標符号量に近い像域フラグの圧縮データを得ることができる。

30

#### 【0279】

<第7の実施形態>

第5の実施形態では、符号化制御部2813と、符号化シーケンス制御部108から、それぞれ第4のカウンタ2811と、第1のカウンタ107からのカウント値を、第6のカウンタ2809へ送り、第6のカウンタ2809では、画像データの符号化、像域データの符号化のカウントを合計し、そのカウント値を、画像・像域符号化制御部2819へ送った。そして、画像・像域符号化制御部2819内のレジスタに保持されている画像データおよび像域データの合計目標値に対して、前記画像データ、像域データの符号量の合計カウント値が、この目標値をオーバーしたかどうか判定し、オーバーした際には、符号化制御部1713、符号化シーケンス制御部108に対して、再符号化処理を行うよう制御信号を出力する構成を説明した。

40

#### 【0280】

第7の実施形態では、第4のカウンタ2811と、第1のカウンタ107と、第6のカウンタ2809との、各々のカウント値を参照し、画像データ、像域データの再符号化処理を独立して行う場合の実施形態を説明する。

#### 【0281】

50

画像・像域符号化制御部 1719 内のレジスタには、画像データおよび像域情報の合計目標値（目標符号量）が保持されており、第 6 のカウンタ 2809 から前記画像データ、像域情報の符号量の合計カウント値が送られてくる。この合計符号量がレジスタ内の目標値をオーバーした際には、画像・像域符号化制御部 2819 は、符号化制御部 2813、符号化シーケンス制御部 108 に対して、再符号化処理を行うよう制御信号を出力する。

#### 【0282】

画像・像域符号化制御部 2819 内の目標値をオーバーしたということは、符号化シーケンス制御部 108、符号化制御部 2813 内の各々レジスタ内に保持されている画像データの目標符号量、像域情報の目標符号量に対し、画像データか像域情報のどちらか少なくとも一方がオーバーしたということ（両方の場合もある）である。

10

#### 【0283】

画像・像域符号化制御部 2819 から再符号化の制御信号を受けた符号化制御部 2813、符号化シーケンス制御部 108 は、第 1 のカウンタ 107、第 4 のカウンタ 2811 のカウント値が、符号化制御部 2813、符号化シーケンス制御部 108 内のレジスタに保持されている、画像データの符号量の目標値、像域データの符号量の目標値と参照し、オーバーしていないか判定し、オーバーしている場合には、先に説明した再符号化処理のための制御信号を出力し、再符号化処理を行う。

#### 【0284】

例えば、符号化制御部 2813 内の目標値に対して、像域データの符号量はオーバーせず、符号化シーケンス制御部 108 内の目標値に対しては、画像データの符号量がオーバーし、その結果、画像・像域符号化制御部 1719 の目標値をオーバーしてしまった場合には、符号化制御部 1713 は像域データに対する再符号化制御信号は出力せず、符号化シーケンス制御部 108 は画像データに対して再符号化処理の制御信号を出力する。

20

#### 【0285】

これによって、画像データ、像域データの合計目標値をオーバーして再符号化処理になった場合でも、その目標値をオーバーした要因として支配的なデータの方だけを再符号化処理を行うことで、各々の目標値をオーバーしていない、例えば、上述の例では像域データに関しては再符号化することが不要になるため、無駄な画質劣化を避けることが可能になる。

#### 【0286】

30

##### <第 8 の実施形態>

第 5 の実施形態では、符号化制御部 2813 と、符号化シーケンス制御部 108 から、それぞれ第 4 のカウンタ 2811 と、第 1 のカウンタ 107 からのカウント値を、第 6 のカウンタ 2809 へ送る。第 6 のカウンタ 2809 では、画像データの符号化、像域情報の符号化の符号量（カウント値）を合計し、その合計値を、画像・像域符号化制御部 1719 へ送った。そして、画像・像域符号化制御部 2819 内のレジスタに保持されている画像データおよび像域情報の合計目標値に対して、画像データ及び像域情報の符号量の合計値が、この目標値をオーバーしたかどうかを判定し、オーバーした際には、符号化部 2813（結果として可逆符号再符号化部 2807 を含む）、符号化シーケンス制御部 108 に対して再符号化処理を行う制御信号を出力する構成を説明した。

40

#### 【0287】

かかる構成において、画像・像域符号化制御部 2819 から符号化制御部 2813、符号化シーケンス制御部 108 に対する再符号化処理の制御信号を交互に出力する構成にしても構わない。

#### 【0288】

画像・像域符号化制御部 2819 内のレジスタに保持されている目標値をオーバーした場合には、先ず、符号化シーケンス制御部 108 に対して、再符号化処理の指示要求を出し、再符号化を行う。そして、この再符号化を行っても、なお像域符号化制御部 1719 内のレジスタに保持されている目標値をオーバーした場合には、次に、符号化制御部 2813（可逆符号化部 2805 及び可逆符号再符号化部 2807）に対して再符号化処理を行

50

うよう要求信号を出力する。

【0289】

以下、目標値をオーバーすると判断した毎に、交互に再符号化の圧縮率を高くするようになる。

【0290】

以上の結果、圧縮率を過剰に高くして符号化することを抑制することができるようになる。

【0291】

以上説明した第1乃至第8の実施形態の処理は、始めに説明した基本部分を含めて、マルチタスクOSを搭載した汎用の情報処理装置（例えばパソコン等）によるコンピュータプログラムによっても実現できる。従って、本発明はかかるコンピュータプログラムにも適用できるものである。この場合、図1や、第1乃至第8の実施形態で示したブロック構成図における各ユニットは、コンピュータプログラムのモジュールもしくは関数プログラムで実現できるのは、当業者であれば容易に理解できよう。従って、本願発明は、コンピュータプログラムに適用しても構わない。

10

【0292】

また、通常、パソコン等の汎用情報処理装置にコンピュータプログラムを導入する場合には、そのコンピュータプログラムを記憶する記憶媒体（フロッピーディスク、CD-ROM、MO等）をセットし、インストール或いはコピーすることで実現できるものであるから、本発明はかかる記憶媒体をもその範疇とするものである。

20

【0293】

【発明の効果】

以上説明したように本発明によれば、多値画像の像域情報を再入力し直すことなく、目的とするサイズに収まる符号化を行うことが可能になる。

【図面の簡単な説明】

【図1】本発明を適用する画像処理装置の第1の基本構成を示す図である。

【図2】本発明を適用する画像処理装置の第2の基本構成を示す図である。

【図3】図1の構成における処理を簡略化して示したフローチャートである。

【図4】初期状態の符号化フェーズにおけるデータフローとメモリ内容を表わす図である。

30

【図5】符号化・再符号化フェーズにおけるデータフローとメモリ内容を表わす図である。

。

【図6】転送フェーズにおけるデータフローとメモリ内容を表わす図である。

【図7】転送フェーズ後の符号化フェーズにおけるデータフローとメモリ内容を表わす図である。

【図8】図1の構成における処理の詳細を示すフローチャートである。

【図9】図1の構成の変形例における符号化・再符号化フェーズにおけるデータフローとメモリ内容を表わす図である。

【図10】図9の変形例における転送フェーズにおけるデータフローとメモリ内容を表わす図である。

40

【図11】図9の変形例における転送フェーズ後の符号化フェーズにおけるデータフローとメモリ内容を表わす図である。

【図12】図2の構成における処理手順を示すフローチャートである。

【図13】図2の構成における、初期状態の符号化フェーズにおけるデータフローとメモリ内容を表わす図である。

【図14】図2に構成における、転送フェーズにおけるデータフローとメモリ内容を表わす図である。

【図15】図2の構成における、符号化・再符号化フェーズにおけるデータフローとメモリ内容を表わす図である。

【図16】図2の構成における、符号化・再符号化フェーズ後の符号化フェーズにおける

50

データフローとメモリ内容を表わす図である。

【図 17】本発明の第1の実施形態における装置のブロック構成図である。

【図 18】第1の実施形態における処理手順を示すフローチャートである。

【図 19】第1の実施形態における可逆符号の符号化データと再符号化後の符号化データを示す図である。

【図 20】第1の実施形態における他の構成を示す図である。

【図 21】第2の実施形態における可逆符号の再符号化後の符号化データと再々符号化後の符号化データを表わす図である。

【図 22】第3の実施形態における装置のブロック構成図である。

【図 23】第3の実施形態における処理手順を示すフローチャートである。 10

【図 24】第3の実施形態における可逆符号の符号化データと再符号化後の符号化データを示す図である。

【図 25】第3の実施形態における他の構成を示す図である。

【図 26】第4の実施形態における可逆符号の再符号化後の符号化データと再々符号化後の符号化データを表わす図である。

【図 27】第3の実施形態における初期状態の符号化フェーズにおけるデータフローとメモリ内容を表わす図である。

【図 28】第5の実施形態における装置のブロック構成図である。

【図 29】第5の実施形態における処理手順を示すフローチャートである。

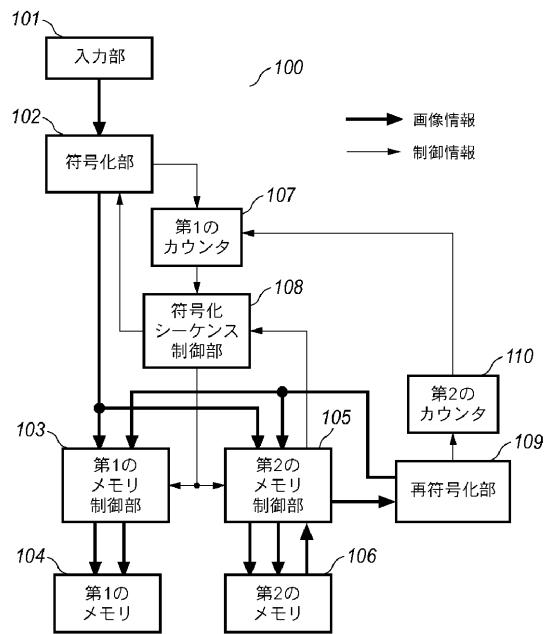
【図 30】第5の実施形態における可逆符号の符号化データと再符号化後の符号化データを示す図である。 20

【図 31】第5の実施形態における他の構成を示す図である。

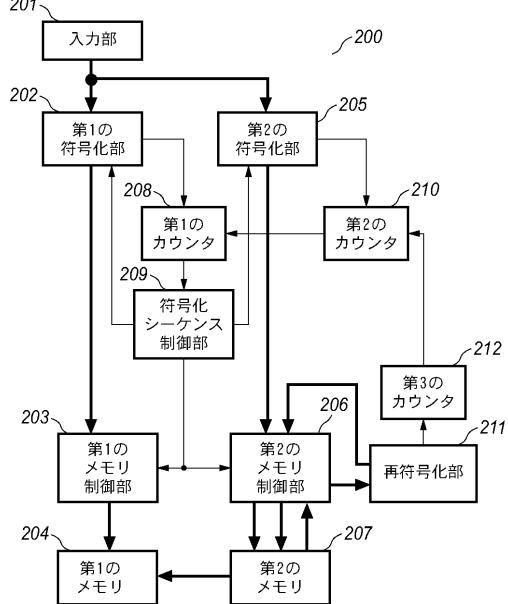
【図 32】第6の実施形態における可逆符号の再符号化後の符号化データと再々符号化後の符号化データを表わす図である。

【図 33】第5の実施形態における初期状態の符号化フェーズにおけるデータフローとメモリ内容を表わす図である。

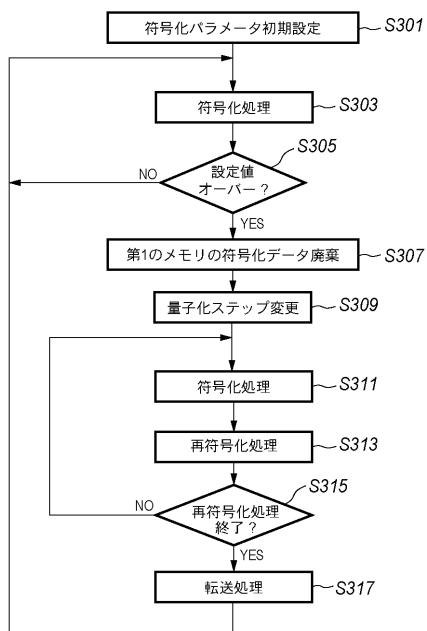
【図1】



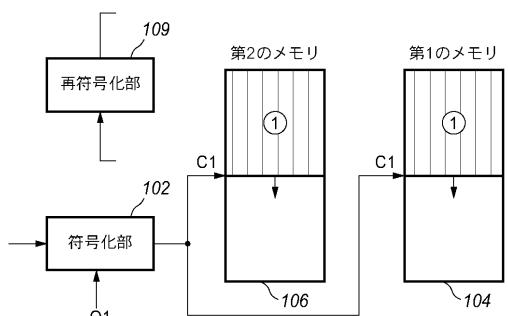
【図2】



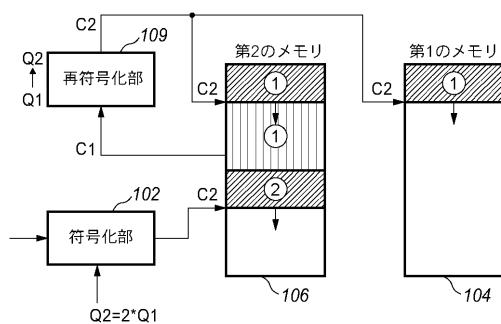
【図3】



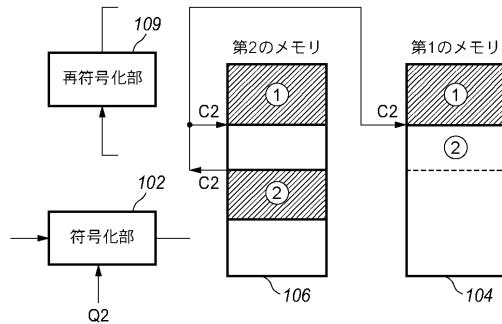
【図4】



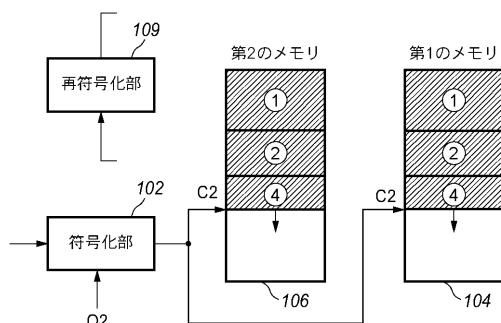
【図5】



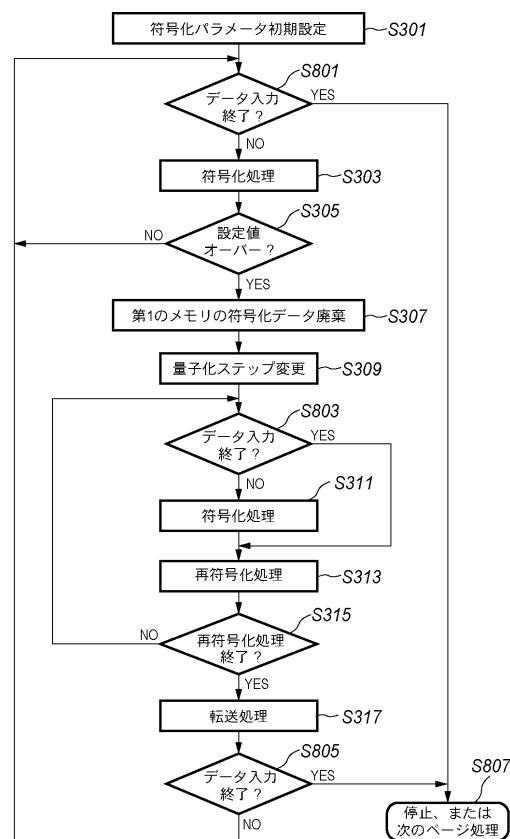
【図6】



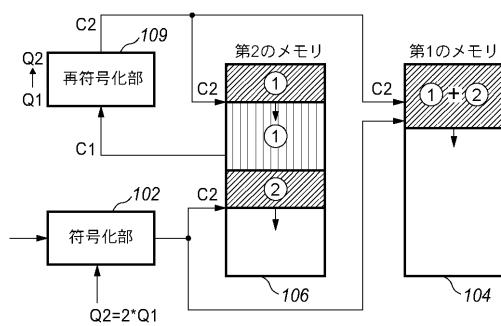
【図7】



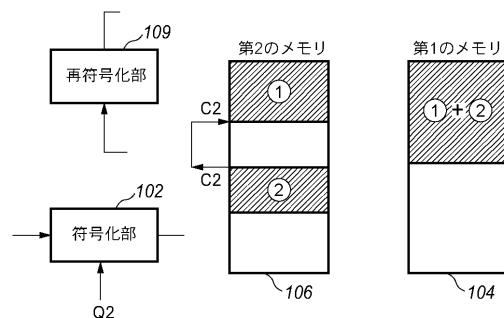
【図8】



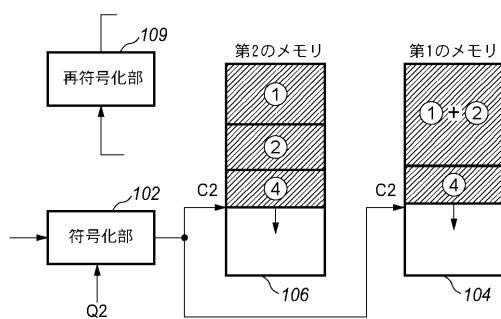
【図9】



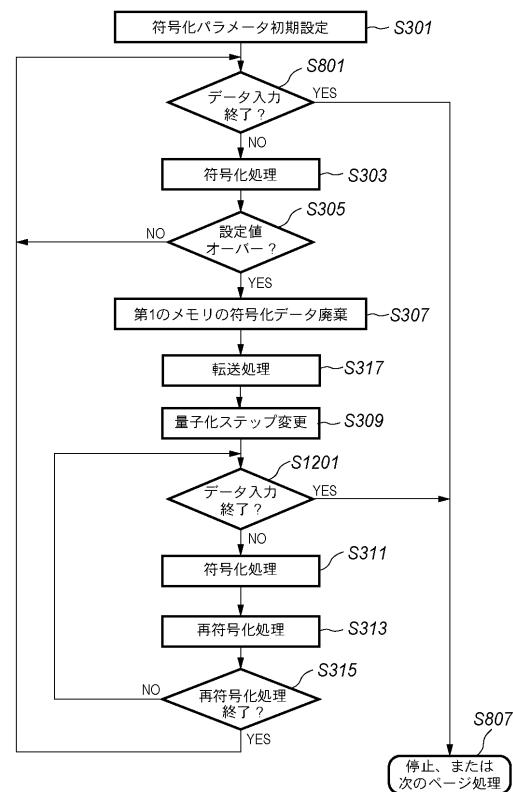
【図10】



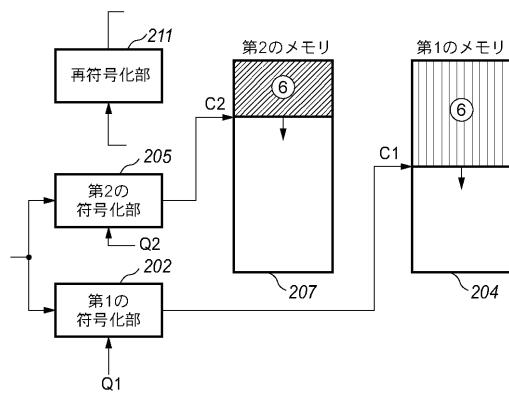
【図11】



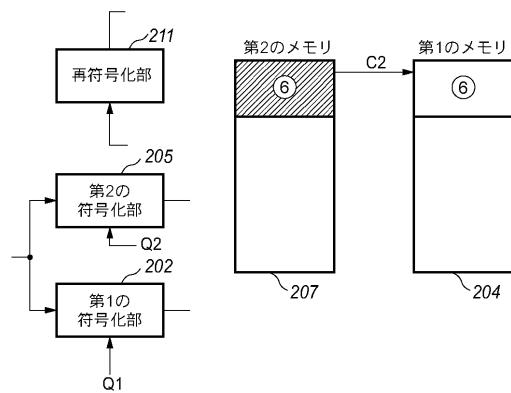
【図12】



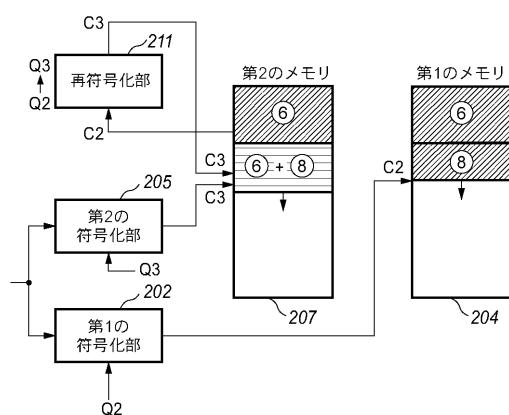
【図13】



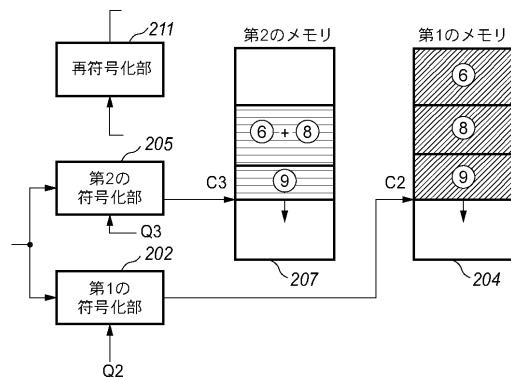
【図14】



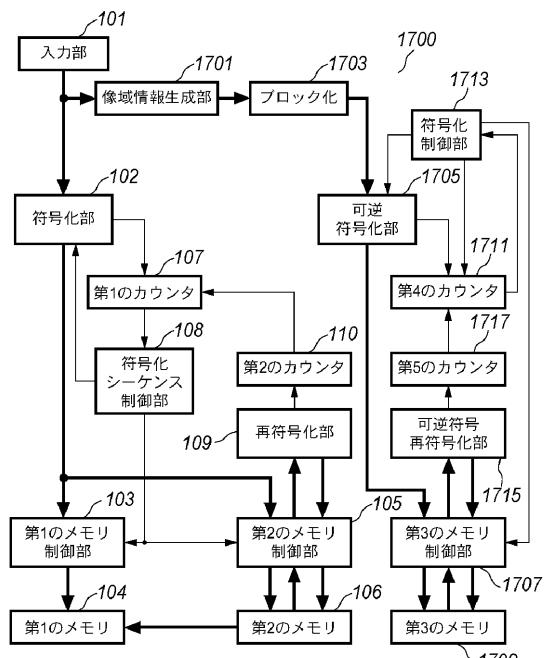
【図15】



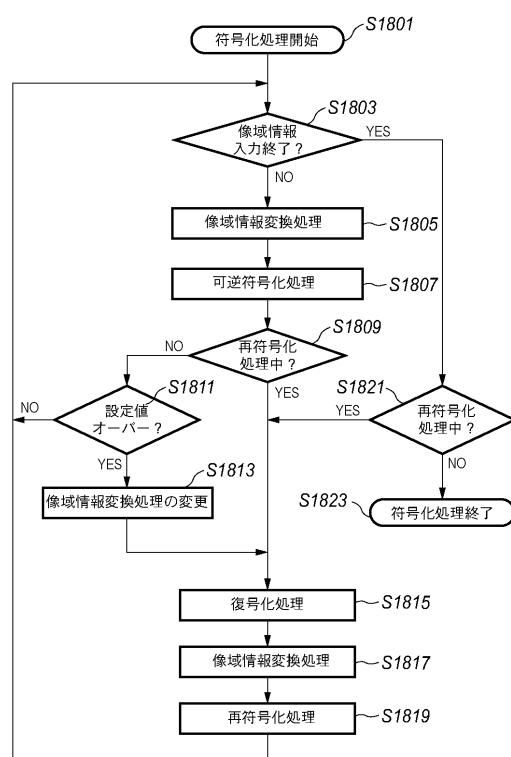
【図16】



【図17】



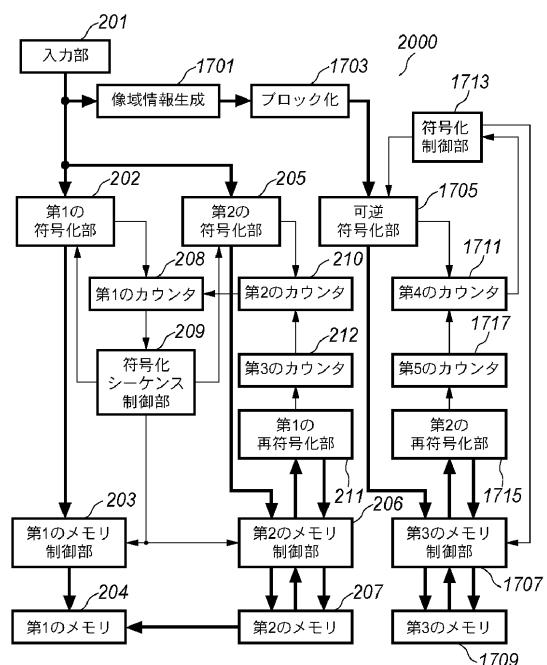
【図18】



【図19】

- (a) 000000BB (8ビット)  
 1/0 文字部／非文字部フラグ  
 有彩色／無彩色フラグ
- (b) 1, 1, 1, 1, 2, 3, 2, 3, 2, 2, 2, 2, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 2,
- (c) -4, 1, 4, 2, 3, 2, 3, -5, 2, 3, 1, 0, 1, -6, 0,
- (d) 1, 1, 1, 1, 3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3,
- (e) -4, 1, -9, 3, -9, 1,

【図20】



【 図 2 1 】

(a) 1, 1, 1, 1, 2, 3, 2, 3, 2, 2, 2, 2, 1, 0, 1, 0, 0, 0, 0, 0, 0, 2,

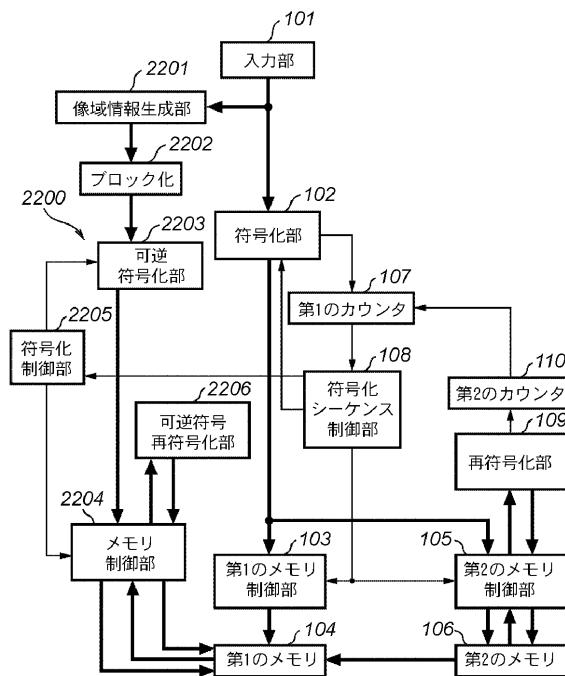
(b) 1, 1, 1, 1, 2, 3, 2, 3, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2,

(c) -4, 1, 4, 2, 3, 2, 3, -5, 2, -9, 0,

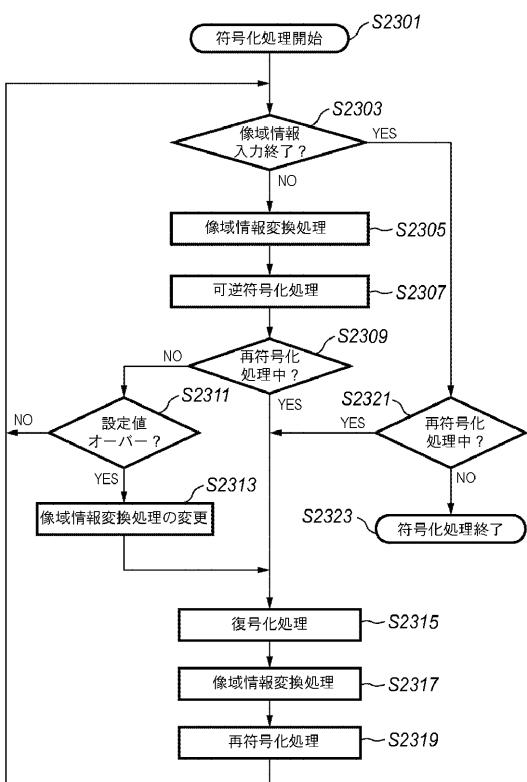
(d) 1, 1, 1, 1, 3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3,

(e) -4, 1, -9, 3, -9, 1,

【 図 2 2 】



【図23】



【図24】



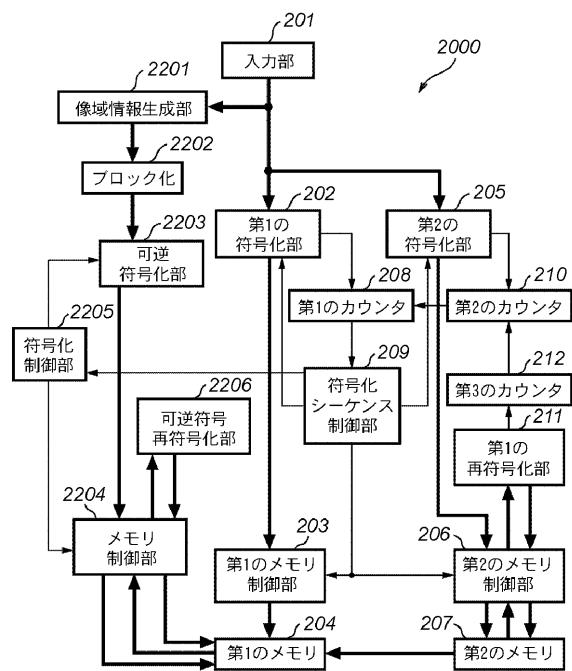
(b) 1, 1, 1, 1, 2, 3, 2, 3, 2, 2, 2, 2, 2, 1, 0, 1, 0, 0, 0, 0, 0, 0, 2,

(c)  $-4, 1, \underline{4}, 2, 3, 2, 3, -5, 2, \underline{3}, 1, 0, 1, -6, 0,$

(d) 1, 1, 1, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3,

$$(e) \quad -4, 1, -9, 3, -9, 1,$$

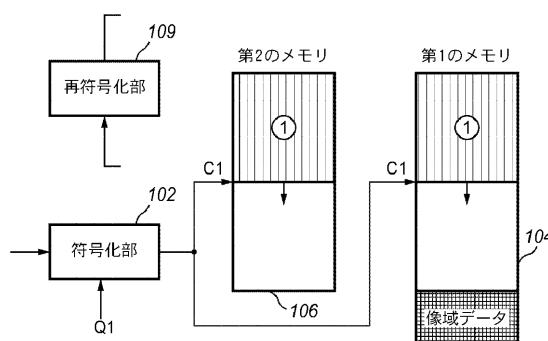
【図25】



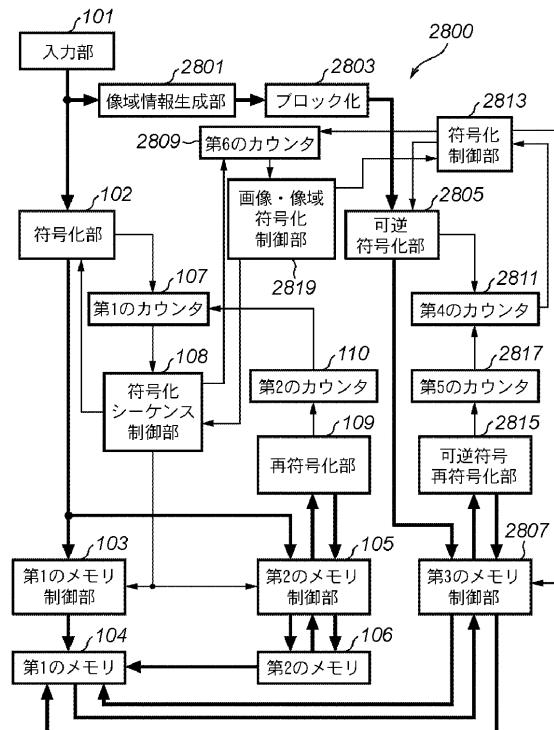
【図26】

- (a) 1, 1, 1, 1, 2, 3, 2, 3, 2, 2, 2, 2, 1, 0, 1, 0, 0, 0, 0, 0, 0, 2,
- (b) 1, 1, 1, 1, 2, 3, 2, 3, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2,
- (c) -4, 1, 2, 3, 2, 3, -5, 2, -9, 0,
- (d) 1, 1, 1, 1, 3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3,
- (e) -4, 1, -9, 3, -9, 1,

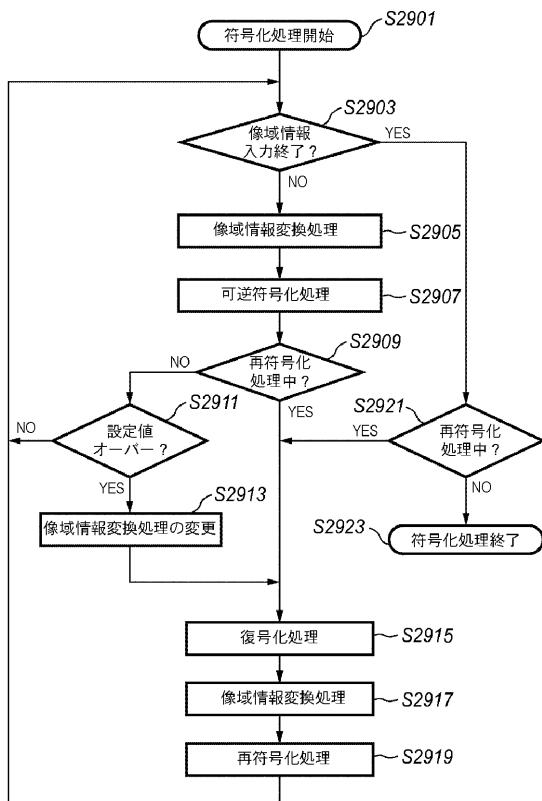
【図27】



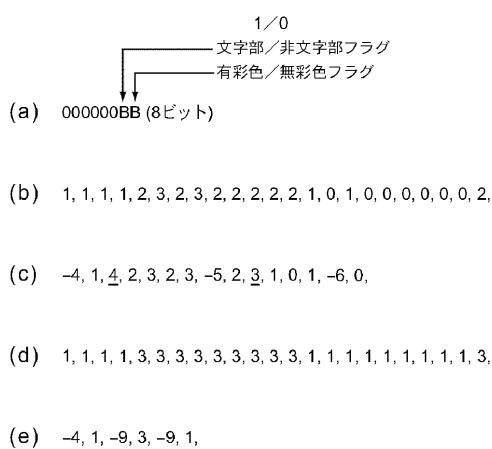
【図28】



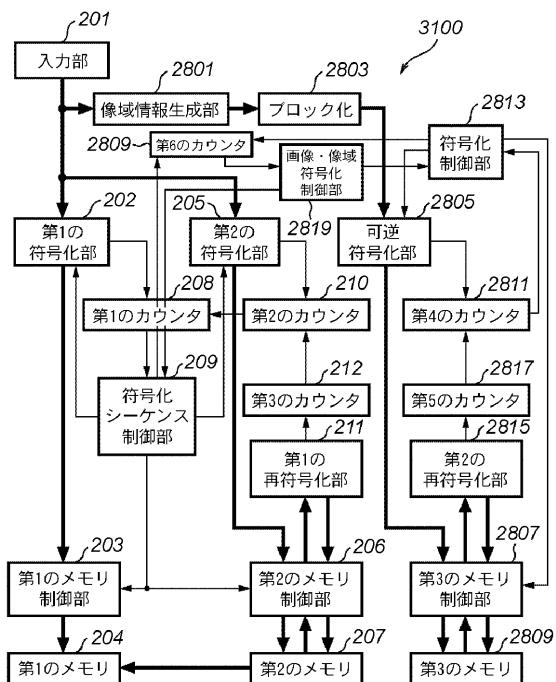
【図29】



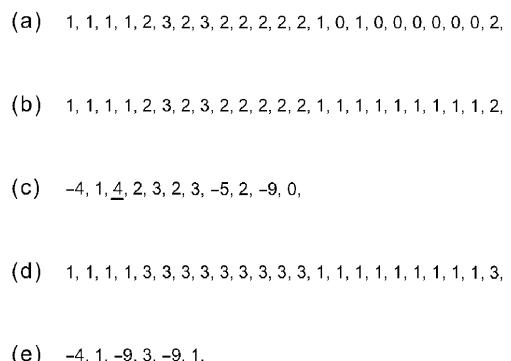
【図30】



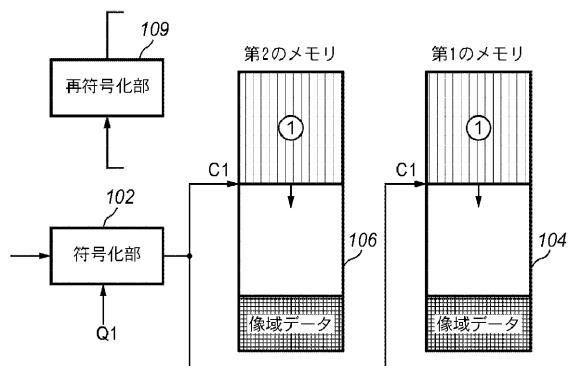
【図31】



【図32】



【図3-3】



---

フロントページの続き

(72)発明者 太田 健一  
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 伊藤 直樹  
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 加藤 進一  
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 大澤 秀史  
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

審査官 松永 稔

(56)参考文献 特開平11-136524(JP,A)  
特開平11-203071(JP,A)

(58)調査した分野(Int.Cl., DB名)

H04N 1/413

H04N 1/40

H04N 7/26