

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第4847398号  
(P4847398)

(45) 発行日 平成23年12月28日(2011.12.28)

(24) 登録日 平成23年10月21日(2011.10.21)

(51) Int.Cl.	F 1
HO4N 1/413 (2006.01)	HO4N 1/413 D
HO4N 7/26 (2006.01)	HO4N 7/13 Z
HO3M 7/30 (2006.01)	HO3M 7/30 Z

請求項の数 14 (全 24 頁)

(21) 出願番号 特願2007-150778 (P2007-150778)  
 (22) 出願日 平成19年6月6日 (2007.6.6)  
 (65) 公開番号 特開2008-306394 (P2008-306394A)  
 (43) 公開日 平成20年12月18日 (2008.12.18)  
 審査請求日 平成22年3月3日 (2010.3.3)

(73) 特許権者 000001007  
 キヤノン株式会社  
 東京都大田区下丸子3丁目30番2号  
 (74) 代理人 100076428  
 弁理士 大塚 康徳  
 (74) 代理人 100112508  
 弁理士 高柳 司郎  
 (74) 代理人 100115071  
 弁理士 大塚 康弘  
 (74) 代理人 100116894  
 弁理士 木村 秀二  
 (74) 代理人 100130409  
 弁理士 下山 治  
 (74) 代理人 100134175  
 弁理士 永川 行光

最終頁に続く

(54) 【発明の名称】 画像処理装置およびその方法

## (57) 【特許請求の範囲】

## 【請求項 1】

前景画素と背景画素からなる画像データを入力する入力手段と、  
 前記画像データにおいて、前記前景画素として出現する画素値のヒストグラムに基づいて、前記背景画素の画素値に前記前景画素として出現しない画素値を設定することにより、第1の画像データを生成する第1の生成手段と、

前記第1の画像データを符号化する第1の符号化手段と、

前記画像データにおいて、前記前景画素の画素値に前記背景画素の画素値に基づいた値を設定することにより、第2の画像データを生成する第2の生成手段と、

前記第2の画像データを符号化する第2の符号化手段とを有することを特徴とする画像処理装置。 10

## 【請求項 2】

前記第1の符号化手段はロスレス符号化を実行し、前記第2の符号化手段はロッサー符号化を実行することを特徴とする請求項1に記載された画像処理装置。

## 【請求項 3】

前記第1の生成手段は、前記背景画素の画素値に、前記前景画素に出現する最小の画素値より一つ小さい値、または、前記前景画素に出現する最大の画素値より一つ大きい値を設定することを特徴とする請求項1に記載された画像処理装置。

## 【請求項 4】

前記第1の生成手段は、前記背景画素の画素値に、前記前景画素に出現する画素値の範

囲の中央値に最も近く、かつ、前記前景画素に出現しない値を設定することを特徴とする請求項1に記載された画像処理装置。

【請求項5】

前記第1の生成手段は、前記背景画素の画素値に、前記前景画素に出現する画素値の中で出現頻度の最も高い輝度値に最も近く、かつ、前記前景画素に出現しない値を設定することを特徴とする請求項1に記載された画像処理装置。

【請求項6】

画像データの注目画素の値と、前記注目画素の複数の周辺画素の値から、前記注目画素を前景画素または背景画素に分類して、前記前景画素と前景画素以外の画素からなる前景画像、および、前記背景画素と背景画素以外の画素からなる背景画像を生成する生成手段と、

前記前景画素として出現する画素値のヒストグラムに基づいて、前記前景画像の前記前景画素以外の画素の値に前記前景画素として出現しない画素値を設定する第1の設定手段と、

前記前景画素以外の画素の値を設定した前景画像をロスレス符号化するロスレス符号化手段と、

前記背景画像の前記背景画素以外の画素の値に背景画素値を設定する第2の設定手段と、

前記背景画素以外の画素の値を設定した背景画像をロッサー符号化するロッサー符号化手段と、

前記ロスレス符号化した前景画像の符号データと、前記ロッサー符号化した背景画像の符号データを合成して、前記画像データの符号データにする合成手段とを有することを特徴とする画像処理装置。

【請求項7】

前記生成手段は、前記注目画素の値と、前記複数の周辺画素の値を比較して、前記注目画素と同値の周辺画素が存在する場合は前記注目画素を前記前景画素に分類にし、前記注目画素と同値の周辺画素が存在しない場合は前記注目画素を前記背景画素に分類することを特徴とする請求項6に記載された画像処理装置。

【請求項8】

前記生成手段は、前記注目画素の値と前記複数の周辺画素の値の差分を計算して、前記差分が所定の閾値に満たない周辺画素が存在する場合は前記注目画素を前記前景画素に分類にし、前記差分が前記所定の閾値に満たない周辺画素が存在しない場合は前記注目画素を前記背景画素に分類することを特徴とする請求項6に記載された画像処理装置。

【請求項9】

前記生成手段は、前記注目画素の値と、前記複数の周辺画素の値を比較する手段と、前記注目画素と同値の周辺画素が存在する場合、前記注目画素と前記複数の周辺画素の間のエッジ強度を算出し、前記エッジ強度が所定の閾値を超える場合は前記注目画素を前記前景画素に分類にし、前記エッジ強度が前記所定の閾値を超えない場合は前記注目画素を前記背景画素に分類する手段と、

前記前景画素に分類された前記注目画素の値によって、前景画素の判定値を更新する手段と、

前記注目画素と同値の周辺画素が存在しない場合、前記注目画素の値が前記判定値に一致する場合は前記注目画素を前記前景画素に分類し、前記注目画素の値が前記判定値に一致しない場合は前記注目画素を背景画素に分類する手段とを有することを特徴とする請求項6に記載された画像処理装置。

【請求項10】

前記第1の設定手段は、色成分ごとに、前記画像データが含む有効な信号レベルの数を計数するカウンタと、

前記色成分ごとに、前記信号レベルの数に基づくインデックステーブルを生成する手段と、

10

20

30

40

50

前記色成分ごとに、前記信号レベルの数が所定の閾値に満たない場合は、前記インデックステーブルに基づき前記前景画素の値をインデックス値に置き換える手段とを有することを特徴とする請求項6に記載された画像処理装置。

【請求項11】

前景画素と背景画素からなる画像データを入力し、

前記画像データにおいて、前記前景画素として出現する画素値のヒストグラムに基づいて、前記背景画素の画素値に前記前景画素として出現しない画素値を設定することにより、第1の画像データを生成し、

前記第1の画像データを符号化し、

前記画像データにおいて、前記前景画素の画素値に前記背景画素の画素値に基づいた値を設定することにより、第2の画像データを生成し、

前記第2の画像データを符号化することを特徴とする画像処理方法。 10

【請求項12】

画像データの注目画素の値と、前記注目画素の複数の周辺画素の値から、前記注目画素を前景画素または背景画素に分類して、前記前景画素と前景画素以外の画素からなる前景画像、および、前記背景画素と背景画素以外の画素からなる背景画像を生成し、

前記前景画素として出現する画素値のヒストグラムに基づいて、前記前景画像の前記前景画素以外の画素の値に前記前景画素として出現しない画素値を設定し、

前記前景画素以外の画素の値を設定した前景画像をロスレス符号化し、

前記背景画像の前記背景画素以外の画素の値に背景画素値を設定し、

前記背景画素以外の画素の値を設定した背景画像をロッキー符号化し、

前記ロスレス符号化した前景画像の符号データと、前記ロッキー符号化した背景画像の符号データを合成して、前記画像データの符号データにすることを特徴とする画像処理方法。 20

【請求項13】

コンピュータ装置を請求項1から請求項10の何れか一項に記載された画像処理装置の各手段として機能させることを特徴とするプログラム。

【請求項14】

請求項13に記載されたプログラムが記録されたことを特徴とするコンピュータが読み取り可能な記録媒体。 30

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、画像データを符号化する画像処理に関する。

【背景技術】

【0002】

文字や線画などの画像（以下、文字画像）と写真画像が混在する画像（以下、混在画像）は、一つの符号化方式で高压縮符号化すると画質劣化が問題になる。

【0003】

混在画像の符号化における画質劣化を防ぐために、例えば、非特許文献1は、次の方法を開示する。まず、文字画像の領域（以下、文字領域）、写真画像の領域（以下、写真領域）、領域を特定する属性データ（二値マスク情報）の3レイヤ構造の画像データにする。そして、各領域に、その領域の特性に合った符号化方式を適用する。 40

【0004】

3レイヤ構造の画像データを利用する方法は、文字領域と写真領域の画像データのほかに、マスク情報を必要とするので、データ量が多くなる問題がある。

【0005】

また、特許文献1は、3レイヤ構造に分けて符号化するのではなく、エッジ情報を用いて文字領域を抽出し、文字領域と文字以外の領域（以下、下地領域）に分離して符号化する方式を開示する。この方法は、文字領域は二値化して二値のロスレス符号化を適用する。 50

また、下地領域には多値画像符号化を適用する。

【0006】

文字領域と下地領域に分離して、文字領域を二値化する方法は、文字画像が階調やグラデーションを有する場合に対応することができない。

【0007】

【特許文献1】特開平8-331391号公報

【非特許文献1】ITU-T Recommendation T.44 | ISO/IEC 16485 : 2000, Information technology - Mixed Raster Content (MRC)

【発明の開示】

【発明が解決しようとする課題】

10

【0008】

本発明は、文字画像と写真画像が混在する画像を、画質劣化を防いで効率的に符号化することを目的とする。

【課題を解決するための手段】

【0009】

本発明は、前記の目的を達成する一手段として、以下の構成を備える。

【0010】

本発明にかかる画像処理は、前景画素と背景画素からなる画像データを入力し、前記画像データにおいて、前記前景画素として出現する画素値のヒストグラムに基づいて、前記背景画素の画素値に前記前景画素として出現しない画素値を設定することにより、第1の画像データを生成し、前記第1の画像データを符号化し、前記画像データにおいて、前記前景画素の画素値に前記背景画素の画素値に基づいた値を設定することにより、第2の画像データを生成し、前記第2の画像データを符号化することを特徴とする。

20

【0011】

また、画像データの注目画素の値と、前記注目画素の複数の周辺画素の値から、前記注目画素を前景画素または背景画素に分類して、前記前景画素と前景画素以外の画素からなる前景画像、および、前記背景画素と背景画素以外の画素からなる背景画像を生成し、前記前景画素として出現する画素値のヒストグラムに基づいて、前記前景画像の前記前景画素以外の画素の値に前記前景画素として出現しない画素値を設定し、前記前景画素以外の画素の値を設定した前景画像をロスレス符号化し、前記背景画像の前記背景画素以外の画素の値に背景画素値を設定し、前記背景画素以外の画素の値を設定した背景画像をロッサー符号化し、前記ロスレス符号化した前景画像の符号データと、前記ロッサー符号化した背景画像の符号データを合成して、前記画像データの符号データにすることを特徴とする。

30

【発明の効果】

【0012】

本発明によれば、文字画像と写真画像が混在する画像を、画質劣化を防いで効率的に符号化することができる。

【発明を実施するための最良の形態】

【0013】

以下、本発明にかかる実施例の画像処理を図面を参照して詳細に説明する。

40

【実施例1】

【0014】

【符号化装置】

図1は実施例の符号化装置の構成例を示すブロック図である。

【0015】

符号化装置への画素データの入力順序は、ラスタスキャン順とし、各画素においてはR、G、Bの順に色成分データが入力されるものとする。R、G、Bそれぞれに成分番号0、1、2を定義し、画像の左上隅の座標を(0, 0)とする。水平方向の画素位置がx、垂直方向の画素位置がyの画素の成分番号Cの値をP(x, y, C)と表すこととする。例えば、位置(3, 4)にある画素が(R, G, B)=(255, 128, 0)という値をもつ場合、P(3, 4, 0)=255、P(3, 4, 1)=

50

128、 $P(3, 4, 2)=0$ と表現する。

【0016】

ラインバッファ101は、信号線108から画像データを入力し、複数ライン分の画像データをバッファする。ここでは、ラインバッファ101がバッファ可能なライン数をThとする。一画素がRGB各8ビット(3バイト)とし、1ラインの画素数がWとすると、ラインバッファは $3 \times W \times Th$ バイトの記憶容量を有する。なお、ラインバッファ101は、原画像のThライン分の帯状の画像をバッファする。この帯状の画像をストライプもしくはストライプ画像と呼ぶことにする。説明の便宜上、原画像の垂直方向の画素数Hは、Thの整数倍とし、画像の末尾で不完全なストライプが発生しないものとする。

【0017】

領域判定部102は、ラインバッファ101にバッファされた複数ラインの画像データについて、注目画素が前景に含まれるか、背景に含まれるかの判定(以下、前景/背景判定)を行う。領域判定部102は、注目画素の各成分について、図2に示す注目画素の周辺の複数の八画素の値a~hの下記に示す成分値をラインバッファ101から読み出す。

```

a = {P(x-1, y-1, 0), P(x-1, y-1, 1), P(x-1, y-1, 2)}
b = {P(x, y-1, 0), P(x, y-1, 1), P(x, y-1, 2)}
c = {P(x+1, y-1, 0), P(x+1, y-1, 1), P(x+1, y-1, 2)}
d = {P(x-1, y, 0), P(x-1, y, 1), P(x-1, y, 2)}
e = {P(x+1, y, 0), P(x+1, y, 1), P(x+1, y, 2)}
f = {P(x-1, y+1, 0), P(x-1, y+1, 1), P(x-1, y+1, 2)}
g = {P(x, y+1, 0), P(x, y+1, 1), P(x, y+1, 2)}
h = {P(x+1, y+1, 0), P(x+1, y+1, 1), P(x+1, y+1, 2)}

```

ここで、 $P(x, y, C)$ は注目画素の成分値

【0018】

なお、画像の最初のラインや最後のライン、各ラインの先頭/末尾などで、周辺八画素が画像外になる場合は、符号側と復号側で共通の成分値、例えば0をもつこととする。

【0019】

領域判定部102は、注目画素の成分値 $P(x, y, C)$ と、周辺八画素の成分値を比較する。そして、注目画素と同一の画素値をもつ近傍画素の存在の有無を示す情報を、前景画像処理部110の透過色設定部103と、背景画像処理部111の背景画素設定部104に出力する。なお、画素値が同一とは、二つの画素の値を $(r1, g1, b1)(r2, g2, b2)$ とする場合、 $r1=r2, g1=g2, b1=b2$ というように、全成分値が同一の場合を指す。

【0020】

領域判定部102は、注目画素の周辺八画素に、注目画素の値Xに一致する値をもつ画素が一画素でも存在する場合、注目画素を前景画素と判定して、透過色設定部103へ出力する。さらに、注目画素が前景画素であることを示す判定情報(例えば‘1’)を透過色設定部103と背景画素設定部105へ出力する。

【0021】

一方、領域判定部102は、注目画素の周辺八画素に、注目画素の値Xと一致する値をもつ画素が一画素も存在しない場合は、注目画素を背景画素と判定して、背景画素設定部105へ出力する。さらに、注目画素が背景画素であることを示す判定情報(例えば‘0’)を透過色設定部103と背景画素設定部105へ出力する。つまり、領域判定部102は、注目画素を前景画素と背景画素に分類する。

【0022】

透過色設定部103は、領域判定部102から入力される注目画素と、判定情報に基づき前景画像生成処理を行う。判定情報が‘1’、つまり前景画素の場合は注目画素の値Xをそのままロスレス符号化部104へ出力し、判定情報が‘0’、つまり背景画素の場合は任意の画素値tをロスレス符号化部104へ出力する。なお、透過色設定部103が出力する画素値を $X_f$ とする。

【0023】

10

20

30

40

50

透過色設定部103が注目画素の値Xを置き換える画素値tを以下では「透過色」と呼ぶ。なお、この透過色としては、前景画素として出現しない画素値が設定される。また、その画素値（透過色）は、前景画素として出現する画素値のヒストグラムに基づいて設定される。

【0024】

透過色tの設定

以下、透過色tの設定方法の具体例を説明する。

【0025】

まず、前景画像として抽出した画素（前景画素）のR、G、Bそれぞれの成分から輝度値（画素値）の出現頻度を調べ、ヒストグラムを生成する。

10

【0026】

次に、そのヒストグラム（画素値の分布状況）に基づいて、後段でロスレス符号化を行う際の符号化効率をより良くするために最適な画素値を透過色として設定する。例えば、その最適な画素値の決め方として、以下の方法を適用することができる。ただし、以下で示す数値はそれぞれの条件の関係を分かり易くするための一例で、その数値に限定されるものではない。

(I) 前景画素が同じような色（画素値）で構成される場合

【0027】

図19Aは、前景画像に存在する輝度値（画素値）の分布が次の条件に合う場合のヒストグラムの代表図である。

20

(1) 1回以上カウントされた輝度値が27個

(2) 非隣接部分（1回以上カウントされた値が隣接する値域（以下、隣接値域）と別の隣接値域との境界）が0個

(3) 隣接値域の幅（隣接値域が複数ある場合は、1回以上カウントされた輝度値の個数が最大の隣接値域）が27になる場合

【0028】

例えばR、G、Bそれぞれヒストグラムが図19Aに示すような特性を示す場合、前景画素が同じような色（画素値）で構成されていると言える。すなわち、図19(a)のヒストグラムは、出現した輝度値（画素値）が一部分に密集した分布を示す。このような場合、透過色tとして、出現した輝度値（画素値）の最小値に隣接する前の値（小さい値）、または、最大値に隣接する後の値（大きい値）を設定する。図19Aの例では、最小輝度値が43、最大輝度値が69であるから、輝度値42か70を透過色tに設定する。

30

(II) 前景画素の輝度値（画素値）はある程度集中するが「まばら」な場合

【0029】

図19Bは、前景画像に存在する輝度値（画素値）の分布の次の条件に合う場合のヒストグラムの代表図である。

(4) 1回以上カウントされた輝度値が2～10個

(5) 非隣接部分（隣接値域と別の隣接値域の境界）が4個

(6) 隣接値域の幅（隣接値域が複数ある場合は、1回以上カウントされた輝度値の個数が最大の隣接値域）が10になる場合

40

【0030】

例えばR、G、Bそれぞれヒストグラムが図19Bに示すような特性を示す場合、前景画素として出現する輝度値（画素値）の分布はある程度集まって（集中して）はいるが、まだ疎らな状態にあると言える。この場合、出現した輝度値の範囲の中心（中央値）を調べ、中央値に一番近い値を透過色tとして設定する。図19Bの例では、出現する最小輝度値が44で最大輝度値が78であるから、中心値は60になる。60に一番近い、出現していない輝度値は58であるから、透過色t=58に設定する。

(III) 前景画素の輝度値（画素値）が分散している場合

【0031】

図19Cは、前景画像に存在する輝度値（画素値）の分布の次の条件に合う場合のヒスト

50

グラムの代表図である。

- (7) 1回以上カウントされた輝度値が1～13個
- (8) 非隣接部分（隣接値域と別の隣接値域の境界）が33個
- (9) 隣接値域の幅（隣接値域が複数ある場合は、1回以上カウントされた輝度値の個数が最大の隣接値域）が13になる場合

【0032】

例えばR、G、Bそれぞれヒストグラムが図19Cに示すような特性を示す場合、前景画素として出現する輝度値（画素値）の分布が集まっていない（分散した）状態にあると言える。この場合、最も出現頻度が高い輝度値を調べ、その輝度値に一番近い値を透過色tとして設定する。図19Cの例では、出現頻度が最も高い輝度値は16になる。16に一番近い、出現していない輝度値21を透過色tに設定する。

10

【0033】

以上の(I)(II)(III)の何れかの設定方法を適用すれば、透過色tとして設定される画素（背景画素の部分）が他の近隣の画素（前景画素の部分）と近い色（画素値）になるので、予測符号化する際は予測誤差を小さく抑えることができる。そして、その予測誤差を可変長符号化する場合、各予測誤差に短い符号語が割り当てられるため、ロスレス符号化の符号化効率が向上することになる。

【0034】

複数の透過色tの設定

図20は輝度値が異なる前景画素が混在する画像例を示す図である。

20

【0035】

図20に示すような画像の場合、ロスレス符号化の符号化効率をさらに向上するには、透過色tを一種類にしない方がよいと考えられる。もし、前景画素として出現しない画素値が2つ以上ある場合は、それら2つ以上の画素値を、透過色tに相当する値として利用することも有効である。すなわち、2以上の数種類の色（画素値）を透過色（t1, t2, ..., tn）として利用すれば符号化効率を向上することができる。

【0036】

以下、複数の透過色tの設定方法の具体例を説明する。

【0037】

透過色tを一種類とする場合と同様、まず、前景画像として抽出した画素（前景画素）のR、G、Bそれぞれの成分から輝度値（画素値）の出現頻度を調べ、ヒストグラムを生成する。

30

【0038】

次に、そのヒストグラム（画素値の分布状況）に基づいて、後段でロスレス符号化を行う際の符号化効率をよりよくするために最適な複数の画素値を透過色（t1, t2, ..., tn）として設定する。例えば、その最適な画素値の決め方として、以下の方法を適用することができる。ただし、以下で示す数値はそれぞれの条件の関係を分かり易くするための一例で、その数値に限定されるものではない。

(IV) 前景画素が幾つかの色（画素値）で構成される場合

【0039】

図19Dは、前景画像に存在する輝度値（画素値）の分布が次の条件に合う場合のヒストグラムの代表図である。

40

- (10) 1回以上カウントされた輝度値が11～31個
- (11) 非隣接部分（隣接値域と別の隣接値域の境界）が2個
- (12) 隣接値域の幅（隣接値域が複数ある場合は、1回以上カウントされた輝度値の個数が最大の隣接値域）が11、21、31になる場合

【0040】

例えばR、G、Bそれぞれヒストグラムが図19Dに示すような特性を示す場合、前景画素として出現する輝度値（画素値）が幾つかの範囲（値域）に集まっている（集中している）状態にあると言える。例えば10～20の範囲s1、105～130の範囲s2、170～190の範囲s3に集

50

中している場合、それぞれの範囲（値域）に対して適切な透過色t1、t2、t3を設定するのが好ましい。その際、各範囲（各値域）に対する透過色の設定方法は、上記の(I)(II)(III)の何れかの方法を適用することができる。

#### 置き換え

##### 【0041】

次に、例えば3つの透過色t1、t2、t3を画像のどの位置の画素と置き換えるかを説明する。

##### 【0042】

まず、前景画像のラインごとにヒストグラムを生成する。ヒストグラムを生成した結果、注目ラインに輝度値が存在しない場合は、透過色t1、t2、t3の何れかをデフォルト値として設定する。本実施例ではt1をデフォルト値とする。 10

##### 【0043】

注目ラインに輝度値が存在する場合は、その輝度値がヒストグラムのどの範囲s1、s2、s3に属すかによって、透過色を設定する。例えば、範囲s1の輝度値しか存在しない場合は、そのラインの前景として抽出された画素（前景画素）以外の画素に対して、透過色t1を設定する。

##### 【0044】

注目ラインに複数の範囲の輝度値が存在する場合は、どちらか一方の輝度値を設定する。例えば、ラスタスキャン順に走査して、先に出現した輝度値が属す範囲によって透過色を設定してもよいし、ライン中のヒストグラムの分布によって、出現頻度の高い輝度値に近い透過色を設定してもよい。 20

##### 【0045】

また、ラインごとに透過色を設定するのではなく、前景画像をn×mのブロックに分割し、ブロック内の輝度値のヒストグラムを生成して、上記と同様の方法により、ブロックごとに透過色を設定してもよい。

##### 【0046】

上記の方法により設定された透過色tはヘッダ情報として符号化データに付加して、復号時に、どの値が透過色tであるかが分かるようにする。

##### 【0047】

透過色tには、前景画素の値として未使用の値を調べて、その値を設定する。なお、存在する可能性が低い画素値を透過色tと予め設定してもよい。言い換えれば、透過色tは、前景画素の値として未使用の値であれば、どのような値でも構わない。 30

##### 【0048】

一方、背景画素設定部105は、領域判定部102から入力される注目画素と、判定情報に基づき背景画像生成処理を行う。判定情報が‘0’、つまり背景画素の場合は注目画素の値Xをそのままロッキー符号化部106へ出力し、判定情報が‘1’、つまり前景画素の場合は、注目画素の値Xとして何らかの背景画素値を割り当て、割り当てた値をロッキー符号化部106へ出力する。なお、背景画素設定部105が出力する画素値をXbとする。

##### 【0049】

背景画素設定部105が前景画素と判定された注目画素に割り当てる背景画素値は、例えば、注目画素の前方の背景画素の値を設定すればよい。また、所定の値を設定してもよいし、注目画素の上方の背景画素の値と左方の背景画素の値の平均値を設定するなど、背景画像の符号化において符号化効率が良好な値を背景画素値に設定すればよい。 40

##### 【0050】

符号データ生成部107は、図3に一例を示す形態で、ロスレス符号化部104から入力される符号データ（以下、ロスレスデータ）と、ロッキー符号化部106から入力される符号データ（以下、ロッキーデータ）を一つの符号データにまとめる。そして、一つにまとめた符号データを信号線109に出力する。

##### 【0051】

図3(a)に示すように、符号データの先頭にはヘッダとして、ロスレスデータとロッキー 50

データの各データ長（サイズ）の付加情報をもつ。そして、ロスレス符号化された前景画像を示す1ビットの情報に続いてロスレスデータが存在する。さらに、ロッサー符号化された背景画像を示す1ビットの情報に続いてロッサーデータが存在する。例えば、前景画像の符号データを示す情報は‘1’、背景画像の符号データを示す情報は‘0’にする。

#### 【0052】

勿論、原画像の全画素が前景画素と判定された場合は、図3(b)に示すように、ロスレスデータサイズを示すヘッダ、前景画像の符号データを示す情報‘1’、ロスレスデータによって構成される符号データが生成される。同様に、原画像の全画素が背景画素と判定された場合は、図3(c)に示すように、ロッサーデータサイズ、背景画像の符号データを示す情報‘0’、ロッサーデータによって構成される符号データが生成される。

10

#### 【0053】

##### ロスレス符号化部

ロスレス符号化部104は、透過色設定部103から入力される画像データをバッファし、画像データをロスレス符号化する。ロスレス符号化部104は、ロスレス符号化を行う符号化器であればどのようなものでも構わない。例えば、連続階調静止画像の可逆的および準可逆的な圧縮方法の国際標準として勧告されるJPEG-LS (ISO/IEC 14495-1および2)などを利用することができる。

#### 【0054】

図4はロスレス符号化部104の構成例を示すブロック図で、画像データを予測誤差に変換する系列変換の処理に周辺画素を用いた予測変換を用い、符号化処理にハフマン符号化を用いる画像圧縮方式の一例を示す。

20

#### 【0055】

ロスレス符号化部104が実際に符号化を行う前に、予め幾つかの画像を示す画像データを系列変換して得られた予測誤差の頻度分布を調べて、図5に示すようなハフマンテーブルを作成し、メモリ1004に格納しておく。予測誤差の頻度分布の一般的な性質として、予測誤差0を中心として出現頻度が高く、予測誤差の絶対値が大きくなるにつれて出現頻度が下がる傾向がある。従って、メモリ1004に格納するハフマンテーブルによって、予測誤差が0近辺のデータには短い符号語が割り当てられ、予測誤差の絶対値が大きいデータには長い符号語が割り当てられる。

#### 【0056】

30

バッファ1001は、透過色設定部103（信号線1006）から画像データを入力する。バッファ1001に1ライン分の画像データが蓄積すると、予測器1002は、バッファ1001から符号化対象画素の直前の画素の画素値dと、1ライン前の画素の画素値bを取り出し、例えば $p=(d+b)/2$ を演算して予測値pを生成する。

#### 【0057】

減算器1003は符号化対象画素の画素値 $X_f$ と予測値pの差分値 $e=X_f-p$ を演算し出力する。ハフマン符号器1005は、メモリ1004に格納されたハフマンテーブルを参照して、差分値e（シンボル）に対応する符号を符号データ生成部107（信号線1007）に出力する。

#### 【0058】

##### ロッサー符号化部

40

ロッサー符号化部106は、背景画素設定部105から入力される画像データをバッファし、画像データをロッサー符号化する。ロッサー符号化部106には、多値の写真画像を対象とする、国際標準として勧告されたJPEG (ITU-T勧告 T.81 | ISO/IEC 10918-1) などに適合または準拠する符号化器を適用することができる。

#### 【0059】

つまり、画素値が同値の近傍画素が存在すれば、注目画素が文字領域を構成する画素である可能性が高い。この性質を利用して、文字領域を構成する画素を前景画素として抽出し分離して、ロスレス符号化する。ただし、この方法によれば、白い下地や同一色の下地も前景画像として抽出し分離することになるが、同一色が連続する画像のロスレス符号化は圧縮効率が高い上、ブロックノイズも発生せず、下地画像の画質劣化を防ぐことができる。

50

る。

#### 【0060】

また、注目画素の値と近傍画素の値が一致する場合に注目画素を前景画素として分離すれば、僅かな色の違いで前景画素として分離されない注目画素が発生する。そこで、注目画素の値と、近傍画素の値の差分diffを次式によって計算し、差分diffが例えば実験的に求めた閾値range未満であれば、注目画素の値と近傍画素の値が類似するとして、注目画素を前景画素として分離してもよい。

$$diff = \{(R_i - R_n)^2 + (G_i - G_n)^2 + (B_i - B_n)^2\}$$

ここで、 $R_i$ 、 $G_i$ 、 $B_i$ は注目画素の色成分値

$R_n$ 、 $G_n$ 、 $B_n$ は近傍画素の色成分値

10

#### 【0061】

##### [復号装置]

次に、上記の符号化装置が生成した符号データを復号する復号装置を説明する。図6は実施例の復号装置の構成例を示すブロック図である。

#### 【0062】

符号バッファ401は、信号線406を介して復号対象の符号データを入力する。ヘッダ解析部402は、符号バッファ401に格納された符号データを読み込み、符号データに付加されたヘッダ情報を解析して、画像データの復号に必要な情報を取り出す。

#### 【0063】

ヘッダ解析部402は、ヘッダ情報に記述されたデータサイズ分のデータを、ヘッダに続く符号データから切り出す。勿論、ヘッダ情報に複数のデータサイズが記述されていれば、対応する符号データをそれぞれ切り出すことは言うまでもない。そして、切り出した符号データの先頭ビットが‘1’であれば当該符号データを前記復号部403へ出力し、先頭ビットが‘0’であれば当該符号データを背景復号部404へ出力する。

20

#### 【0064】

前景復号部403は、ヘッダ解析部402から入力されるロスレスデータを復号して、復号した画像データを画像合成部405へ出力する。同様に、背景復号部404は、ヘッダ解析部402から入力されるロッシャーデータを復号して、復号した画像データを画像合成部405へ出力する。

#### 【0065】

30

画像合成部405は、前景復号部403から入力される画像データ（以下、前景データ）と、背景復号部404から入力される画像データ（以下、背景データ）を合成して、合成結果の画像データ、言い換えれば復号された画像データを信号線407に出力する。

#### 【0066】

図7は画像合成部405の処理を示すフローチャートである。

#### 【0067】

画像合成部405は、復号が開始されると、注目画素の垂直方向の位置をカウントするカウンタ $y$ と、水平方向の位置をカウントするカウンタ $x$ を1に初期化する（S501）。

#### 【0068】

次に、画像合成部405は、前景データの注目画素の値 $X_f(x, y)$ と透過色 $t$ を比較する（S502）。そして、 $X_f(x, y) = t$ の場合は、注目画素の値 $X_f(x, y)$ をそのまま出力する（S503）。また、 $X_f(x, y) \neq t$ の場合は、背景データの注目画素の値 $X_b(x, y)$ を読み出し（S504）、値 $X_b$ を出力する（S505）。なお、透過色 $t$ は、ヘッダ情報の解析結果に基づきヘッダ解析部402から提供される。

40

#### 【0069】

次に、画像合成部405は、カウンタ $x$ をインクリメントし（S506）、カウンタ $x$ と水平方向の画素数 $W$ を比較する（S507）。そして、 $x < W$ の場合は処理をステップS502に戻して注目画素を一つ進める。なお、画素数 $W$ は、ヘッダ情報の解析結果に基づきヘッダ解析部402から提供される。

#### 【0070】

50

x=Wの場合、画像合成部405は、カウンタxを1にし、カウンタyをインクリメントし(S508)、カウンタyと垂直方向の画素数Hを比較する(S509)。そして、y < Hの場合は処理をステップS502に戻して注目ラインを一つ進める。なお、画素数Hは、ヘッダ情報の解析結果に基づきヘッダ解析部402から提供される。

【0071】

y=Hの場合、画像合成部405は、処理を終了する。

【0072】

このように、原画像の注目画素の値と、その周辺八画素の値を比較して、画素値が一致(または類似)する近傍画素が存在するか否かにより前景/背景判定を行い、判定結果に基づき画素を前景画素と背景画素を分離する。そして、画素の連結性が高い前景画素からなる前景画像をロスレス符号化する。また、画素の連結性が低い背景画素からなる背景画像をロッキー符号化する。さらに、前景画像の背景画素位置の画素値を透過色tに置き換えてロスレス符号化する。

10

【0073】

従って、透過色tへの置き換えによってマスク情報を必要とせずに、文字領域と写真領域が混在する原画像を効率的に圧縮符号化することができる。また、階調やグラデーションを有する文字画像も前景画像として分離することができるので、文字画像の階調やグラデーションを維持して、原画像を圧縮符号化することができる。

【0074】

[装置の構成例]

20

コンピュータ装置に符号化プログラムおよび復号プログラムを実行させることで、上記の符号化装置および復号装置は実現可能である。図8はコンピュータ装置の構成例を示すプロック図である。

【0075】

CPU 1401は、RAM 1402をワークメモリとして、ROM 1403やハードディスクドライブ(HDD) 1407に格納されたオペレーティングシステム(OS)やプログラムに従い、システムバス1410を介して、各構成を制御する。HDD 1407が格納するプログラムには、実施例の符号化プログラムおよび復号プログラムが含まれ、上述した画像データの符号化および復号が可能である。

【0076】

30

CPU 101は、RAM 102に前述したラインバッファ101、バッファ1001、符号バッファ401などを割り当てる。また、CPU 101は、例えばHDD 1407に格納されたハフマンテーブルを読み出し、RAM 102の所定領域に格納して符号化に使用する。

【0077】

記憶媒体ドライブ1408は、光ディスクなどの記憶媒体をドライブする。記憶媒体には、実施例の符号化プログラムや復号プログラムを記録することができる。CPU 101は、記憶媒体からRAM 102にロードした符号化プログラムや復号プログラムを実行して、上述した画像データの符号化および復号を行うこともできる。

【0078】

ユーザは、USBインターフェイス(I/F) 1409に接続されたキーボード1404やマウス1405を利用して符号化または復号プログラムの実行を指示する。当該指示を受信したCPU 101は、符号化または復号プログラムを実行して、ビデオI/F 1412に接続されたモニタ1406にユーザインターフェイスを表示する。そして、符号化または復号するデータの選択または指定、並びに、符号化または復号したデータの出力先を受け付ける。

40

【0079】

ユーザは、HDD 1407や記憶媒体に格納されたデータ、あるいは、外部装置1411または図示しないネットワーク上のサーバ装置上のデータを符号化または復号するデータとして選択または指定可能である。また、HDD 1407、記憶媒体、外部装置1411、ネットワーク上のサーバ装置などを符号化または復号したデータの出力先として選択または指定可能である。あるいは、ユーザは、モニタ1406を出力先に選択して、復号された画像を表示させても

50

よい。

【0080】

CPU 101は、ユーザの指示に従い、選択または指定されたデータを符号化または復号して、符号化または復号したデータを選択または指定された出力先に出力する。

【実施例2】

【0081】

以下、本発明にかかる実施例2の画像処理を説明する。なお、実施例2において、実施例1と略同様の構成については、同一符号を付して、その詳細説明を省略する。

【0082】

実施例1では、注目画素の値Xと周辺八画素の値を比較して、画素値が一致（または類似）する近傍画素があるか否かで前景/背景判定を行う例を説明した。他の方法でも前景/背景判定を行うことができる。例えば、周辺画素のエッジ強度を参照したり、直前の前景画素の値を参照するなどして、前景/背景判定を行ってもよい。実施例2では、前景/背景判定の条件に、周辺画素のエッジ強度と直前の前景画素の情報を加える。これにより、より画像の特性に合った前景/背景判定が可能になる。 10

【0083】

図9は実施例2の領域判定部102の処理を示すフローチャートである。

【0084】

領域判定部102は、符号化が開始されると、注目画素の垂直方向の位置をカウントするカウンタyと、水平方向の位置をカウントするカウンタxを1に初期化し、前景画素の判定値q（直前の前景画素の値）を0に初期化する（S701）。 20

【0085】

次に、領域判定部102は、注目画素の値X(x, y)と周辺八画素の値a～hを比較する（S705）。注目画素の値X(x, y)と周辺八画素の値a～hの少なくとも一つが一致（または類似）する場合は、注目画素のエッジ強度を算出する（S710）。エッジ強度ESは以下の方法で算出する。

【0086】

注目画素の値X(x, y)と周辺八画素の値a～hそれぞれを比較して、値X(x, y)と異なる周辺八画素の値の差分値Dを計算する。なお、差分値Dは、RGB各成分同士の差分絶対値の中で一番大きい差分絶対値を採用する。従って、差分値Dが取り得る範囲は0から255である。 30

【0087】

例えば、値X(x, y)=(255, 128, 0)、並びに、値X(x, y)に一致しない、ある近傍画素の値がXa=(254, 200, 200)だとすると、各成分の差分絶対値はd=(1, 72, 200)である。従って、値X(x, y)と値Xaの差分値はD=200になる。値Xと異なる値をもつ近傍画素が一つの場合は、その差分値をエッジ強度Dにする。また、値Xと異なる値をもつ近傍画素が複数存在する場合は、それらの差分値Dの平均値をエッジ強度ESとする。また、閾値TFBの設定方法は、本実施例では任意の値を指定しておくことにする。なお、自動的に可変の閾値TFBの設定をする場合は、注目画素を中心とする3×3の範囲内で最大の輝度値と最小の輝度値の差分を算出し、その差分の絶対値を3で割った値を閾値TFBとする方法などが考えられる。 40

【0088】

次に、領域判定部102は、エッジ強度ESと所定の閾値を比較する（S711）（ここでは、閾値TFBを50と指定しておく）。そして、ES > TFBならば注目画素は前景画素として、q=X(x, y)により前景画素の判定値を更新する（S712）。そして、判定情報‘1’を出力し（S703）、処理をステップS706に進める。また、ES < TFBならば注目画素は背景画素として判定情報‘0’を出力し（S705）、処理をステップS706に進める。

【0089】

一方、領域判定部102は、注目画素の値X(x, y)と周辺八画素の値a～hが一致（または類似）しない場合は、値X(x, y)と判定値qが一致するか否かを判定する。X(x, y)=qの場合は、注目画素は前景画素として判定情報‘1’を出力する（S703）。また、X(x, y) ≠ qの場 50

合は、注目画素は背景画素として判定情報‘0’を出力する(S705)。

【0090】

次に、領域判定部102は、カウンタ $x$ をインクリメントし(S706)、カウンタ $x$ と水平方向の画素数 $W$ を比較する(S707)。そして、 $x < W$ の場合は処理をステップS702に戻して注目画素を一つ進める。なお、画素数 $W$ は、原画像のヘッダ情報から取得する。

【0091】

$x=W$ の場合、領域判定部102は、カウンタ $x$ を1にし、カウンタ $y$ をインクリメントし(S708)、カウンタ $y$ と垂直方向の画素数 $H$ を比較する(S709)。そして、 $y < H$ の場合は処理をステップS502に戻して注目ラインを一つ進める。なお、画素数 $H$ は、原画像のヘッダ情報から取得する。

10

【0092】

$y=H$ の場合、領域判定部102は、処理を終了する。

【0093】

このように、注目画素の値 $X$ と、周辺八画素の値の比較だけではなく、注目画素のエッジ強度を参照して前景/背景判定を行う。従って、文字領域と写真領域を分離する精度を向上させることができる。

【0094】

さらに、注目画素の値 $X$ と前景画素の判定値 $q$ (直前の前景画素の値)を比較する。その結果、図10に示すような注目画素の値 $X$ と同一の値をもつ画素が周辺八画素に存在しない場合でも、注目画素を前景画素として分離することができる。従って、文字領域の画素に高い確率でロスレス符号化を適用することができ、符号化圧縮による文字画像の画質劣化を抑制することができる。

20

【実施例3】

【0095】

以下、本発明にかかる実施例3の画像処理を説明する。なお、実施例3において、実施例1、2と略同様の構成については、同一符号を付して、その詳細説明を省略する。

【0096】

実施例1では、注目画素の値 $X$ と周辺八画素の値を比較して、画素値が一致(または類似)する近傍画素があるか否かで前景/背景判定を行う例を説明した。他の方法でも前景/背景判定を行うことができる。周辺八画素を超えて画素値を参照する範囲を広げ、例えば、図11に示す太枠で示す注目画素の周辺24画素の値を参照して前景/背景判定を行ってもよい。これにより、より画像の特性に合った前景/背景判定が可能になる。

30

【0097】

実施例3の領域判定部102は、実施例1と同様に、まず注目画素の値 $X$ と周辺八画素の値を比較し前景/背景判定を行う。

【0098】

周辺八画素に注目画素の値 $X$ と同一の(または類似する)画素値をもつ近傍画素が存在しない場合、領域判定部102は、周辺八画素の外周の16画素の値と注目画素の値 $X$ を比較する。この場合も、値 $X$ と同一の(または類似する)画素値をもつ画素が一つでも存在すれば‘1’を、一つも存在しなければ‘0’の判定情報を出力する。

40

【0099】

実施例3によれば、図10に示すような周辺八画素に注目画素の値 $X$ と同一の(または類似する)値をもつ画素が存在しない場合でも、注目画素を前景画素として分離することができる。従って、文字領域の画素に高い確率でロスレス符号化を適用することができ、符号化圧縮による文字画像の画質劣化を抑制することができる

【実施例4】

【0100】

以下、本発明にかかる実施例4の画像処理を説明する。なお、実施例4において、実施例1~3と略同様の構成については、同一符号を付して、その詳細説明を省略する。

【0101】

50

実施例1では、原画像全体を一括して処理する方法を説明した。実施例4では、原画像を矩形ブロックに分割して、ブロックごとに処理する方法を説明する。原画像をブロックに分割して前景/背景判定を行えば、前景画素の色が限定される可能性が高くなり、ロスレス符号化の効率が向上すると考えられる。

#### 【0102】

そのため、実施例4では、前景画素の色が限定色（例えば32色以下）の場合は、画素値をインデックス値に変換してからロスレス符号化を行い、色数が所定数以上の場合は、これまでどおり、画素値をそのままロスレス符号化する方法を説明する。なお、画素値をそのままロスレス符号化する場合の処理は実施例1～3と同様の構成になるため、前景画素が特定の色数以内の場合の処理のみ詳細に説明する。

10

#### 【0103】

##### [符号化装置]

図12は実施例3の符号化装置の構成例を示すブロック図で、図1に示す符号化装置にブロック分割部1201、タイルバッファ1202、前景画素変換部1203を追加し、透過色設定部103を除いた構成である。

#### 【0104】

ブロック分割部1201は、ラインバッファ101から読み込んだストライプ画像を水平画素数Tw、垂直画素数Thの矩形ブロック（タイル）に分割して、タイルバッファ1202へ出力する。タイルバッファ1202は、1タイル分の画像データをバッファする。タイルバッファ1202に必要な最低の記憶容量はTw×Th×3バイトである。

20

#### 【0105】

タイルバッファ1202がバッファするタイルの画像データの水平方向位置x、垂直方向位置yの画素の成分値をP(x, y, c)と定義する。ただし、0 ≤ x ≤ Tw-1、0 ≤ y ≤ Th-1であり、cはR、G、Bの何れかである。

#### 【0106】

領域判定部102は、タイルバッファ1202からタイルの画像データを入力する。前景画素変換部1203は、領域判定部102から出力されるタイルデータと、前景/背景判定の結果を示す判定情報を受信し、画素値をインデックス値に変換する。

#### 【0107】

##### 前景画素変換部

30

図13は前景画素変換部1203の構成例を示すブロック図である。

#### 【0108】

有効レベル数カウンタ1301は、領域判定部102からタイルデータを読み出し、タイルデータの色成分ごとに信号レベル（輝度レベル）の数を計数（カウント）する。言い換えれば、有効レベル数カウンタ1301は、色成分ごとに、何種類の信号レベルがタイルデータに含まれるかをカウントする。有効レベル数カウンタ1301のカウント値である、各色成分の信号レベルの数を有効レベル数と呼び、各色成分の有効レベル数をそれぞれNLV(R)、NLV(G)、NLV(B)と表す。

#### 【0109】

図14は有効レベル数カウンタ1301が有効レベル数をカウントするための二値配列F(c, i)を示す図である。cは成分を表し、R、G、Bの何れかである。iは輝度値を表し、0～255の何れかの整数値である。F(c, i)='0'は注目タイルの成分cに輝度値iが出現しないことを意味し、F(c, i)='1'は注目タイルの成分cに輝度値iが出現することを意味する。

40

#### 【0110】

##### 有効レベル数カウンタの処理

図15は有効レベル数カウンタ1301の処理を示すフローチャートである。

#### 【0111】

有効レベル数カウンタ1301は、図14に示す配列Fの全要素を‘0’に初期化する(S1601)。そして、タイル内の垂直方向の画素位置を示すカウンタyを1に初期化し(S1602)、タイル内の水平方向の画素位置を示すカウンタxを1に初期化する(S1603)。

50

## 【0112】

次に、有効レベル数カウント1301は、タイルデータP(x, y, c)を取得して、配列Fの要素F(c, P(x, y, c))をインクリメントする(S1604)。つまり、P(x, y, R)、P(x, y, G)、P(x, y, B)に対応する配列Fの要素F(R, P(x, y, R))、F(G, P(x, y, G))、F(B, P(x, y, B))を‘1’にする。

## 【0113】

次に、有効レベル数カウント1301は、カウントxをインクリメントし(S1605)、カウントxとタイルの水平方向の画素数Twを比較する(S1606)。そして、 $x < Tw$ の場合は処理をステップS1604に戻し、次の画素を処理する。

## 【0114】

$x=Tw$ の場合、有効レベル数カウント1301は、カウントyをインクリメントし(S1607)、カウントyとタイルの垂直方向の画素数Thを比較する(S1608)。そして、 $y < Th$ の場合は処理をステップS1603に戻し、次のラインを処理する。

## 【0115】

$y=Th$ の場合、有効レベル数カウント1301は、配列Fを調べて、各成分の有効レベル数NLV(c)をカウントする(S1609)。つまり、R成分について、要素F(R, 0)からF(R, 255)のうち値が‘1’の要素の数をカウントしてNLV(R)にする。同様に、G成分、B成分についてNLV(G)、NLV(B)を得る。

## 【0116】

インデックステーブル生成部1302は、有効レベル数カウント1301が保持する配列Fを参照して、図16に示すようなインデックス変換用のテーブル（以下、インデックステーブル）IDXを生成する。なお、インデックステーブルIDXは、インデックス変換部1304が輝度値をインデックス値に変換する際に参照するもので、iは輝度値を表し、0から255の何れかの整数値である。そして、生成したインデックステーブルIDXをインデックステーブルメモリ1303に格納する。さらに、符号データに付加する変換テーブル情報をロスレス符号化部104に出力する。

## 【0117】

## インデックステーブル生成部の処理

図17はインデックステーブル生成部1302の処理を示すフローチャートである。なお、インデックステーブル生成部1302は、色成分ごとにインデックステーブルを生成するが、処理は同一であるので、ここでは成分をcとして説明する。

## 【0118】

インデックステーブル生成部1302は、インデックステーブルメモリ1303が保持するインデックステーブルIDXの全要素を-1に初期化する(S1801)。そして、輝度値を表すカウントiを0に初期化し(S1802)、インデックス値を表す変数idx(c)を0に初期化する(S1803)。

## 【0119】

次に、インデックステーブル生成部1302は、有効レベル数カウント部1301が保持する配列Fを参照して、注目輝度値iに対応するF(c, i)を判定する(S1804)。F(c, i)=‘0’の場合は処理をステップS1807へ進める。

## 【0120】

$F(c, i)=‘1’$ の場合、インデックステーブル生成部1302は、IDX(c, i)にidx(c)を設定するとともに、idx(c)をロスレス符号化部104に出力する(S1805)。

## 【0121】

次に、インデックステーブル生成部1302は、idx(c)をインクリメントし(S1806)、カウントiをインクリメントして(S1807)、 $i < 256$ か否かを判定する(S1808)。 $i < 256$ の場合は処理をステップS1804に戻して、次の輝度値を処理する。

## 【0122】

$i=256$ の場合、インデックステーブル生成部1302は、一つの色成分の付加情報の終了を表す終了コードとして-1をロスレス符号化部104に出力する(S1809)。

## 【0123】

10

20

30

40

50

インデックステーブル生成部1302は、以上の処理をRGB各成分について行い、インデックステーブルメモリ1303にインデックステーブルIDXを生成するとともに、変換テーブル情報をロスレス符号化部104に出力する。

【0124】

インデックス変換部以降の処理

注目タイルの有効レベル数の計数と、インデックステーブルIDXの生成が終了すると、インデックス変換部1304、セレクタ1305、透過色設定部103、ロスレス符号化部104は、タイルバッファ1202に格納されたタイルデータの符号化を開始する。

【0125】

符号化装置は、タイルデータを色成分順、かつ、ラスタスキャン順に読み出して符号化する。つまり、タイルのR成分を符号化し、G成分を符号化し、B成分を符号化する。各成分の符号化処理は共通であるから、色成分をcとして符号化処理を説明する。 10

【0126】

インデックス変換部1304は、インデックステーブルIDXを参照して、タイルバッファ1202から成分順、ラスタスキャン順に読み出された前景画素の輝度値 $P(x, y, c)$ をインデックス値 $IDX(c, P(x, y, c))$ に置き換える。また、インデックス変換部1304は、インデックステーブルIDXに定義されていないインデックス値、つまり要素が-1のインデックス値 $IDX(c, j)$ を一つ選んで透過色tとして、背景画素の輝度値 $P(x, y, c)$ を透過色tで置き換える。

【0127】

セレクタ1305は、タイルバッファ1202から成分順、ラスタスキャン順に読み出された輝度値 $P(x, y, c)$ を入力する。また、インデックス変換部1304が出力するインデックス値 $IDX(c, P(x, y, c))$ または透過色 $t=IDX(c, j)$ を入力する。セレクタ1305は、所定の閾値MLと、有効レベル数カウンタ1301が保持する有効レベル数 $NLV(c)$ を比較して、 $NLV(c) < ML$ の場合はインデックス変換部1304の出力値を選択的に出力する。また、 $NLV(c) \geq ML$ の場合はタイルバッファ1202から読み出された輝度値 $P(x, y, c)$ または透過色tを選択的に出力する。 20

【0128】

ロスレス符号化部104は、セレクタ1305から入力される画像データをロスレス符号化して、後述する有効レベル数 $NLV(c)$ の判定結果を示す情報と変換テーブル情報を含むヘッダを附加した、タイル単位の符号データを出力する。 30

【0129】

図18はタイル単位の符号データの構成例を示す図である。

【0130】

タイル単位の符号データの先頭には、1バイトの有効レベル数 $NLV(c)$ の判定結果を示す情報が附加される。図18(a)に示すように、最上位ビット(MSB)をビット7、最下位ビット LSB)をビット0とする。この場合、ビット2が‘1’の場合は $NLV(R) < ML$ を表し、ビット1が‘1’の場合は $NLV(G) < ML$ を表し、ビット0が‘1’の場合は $NLV(B) < ML$ を表す。なお、ビット7からビット3には‘0’を設定する。

【0131】

$NLV(R)$ 、 $NLV(G)$ 、 $NLV(B)$ の何れかがML未満の場合、つまり、先頭バイト>0の場合は、図18(a)に示すように、符号データの先頭バイトに続いて変換テーブル情報が附加される。もし、 $NLV(R)$ 、 $NLV(G)$ 、 $NLV(B)$ のすべてがML以上の場合、つまり、先頭バイト=0の場合は、図18(b)に示すように、変換テーブル情報は附加されない。また、変換テーブル情報は、 $NLV(c) < ML$ である色成分についてのみ符号データに附加される。 40

【0132】

あるタイルについて、変換テーブル情報 $idx(c)$ が下のようになつたとする。なお、-1は色成分ごとの変換テーブル情報のデリミタである。

0, 1, 2, 3, 4, ..., 254, 255, -1, 0, 128, 255, -1, 0, 64, 128, 192, 255

【0133】

また、 $NLV(R)=255$ 、 $NLV(G)=3$ 、 $NLV(B)=5$ であり、 $ML=32$ の場合、符号データの先頭バイトは‘00000011’=3である。この場合、デリミタの-1によって、変換テーブル情報から色成分GとBの変換テーブル情報を切り出し、符号データに付加する。従って、上の例において、符号データに付加される変換テーブル情報は下のようになる。

0, 128, 255, 0, 0, 64, 128, 192, 255

#### 【0134】

変換テーブル情報を符号データに付加する際、デリミタの-1は0に置き換えられ、変換テーブル情報の末尾にある0は削除される。復号装置は、変換テーブル情報を左から右に順に解析して、前の値よりも小さい、または、等しい値をデリミタと判断する。

#### 【0135】

このように、タイル単位に符号化を行い、必要な変換テーブル情報を各タイルの符号データに含めることによって、復号装置は、タイル単位に画像データを復号することができる。

#### 【0136】

なお、符号データ中に所定値が発生しないように工夫を加えて、特殊なマーカを設定し、各タイルの符号データの先頭または末尾にマーカを配置、または、各タイルの符号データ長を別途管理すれば、タイル単位にランダムアクセスして復号することができる。

#### 【0137】

このように、タイル単位のロスレス符号化が可能になる上、タイルの輝度値の有効レベル数が閾値 $ML$ 未満の場合は、前景画像の当該色成分の輝度値をインデックス値に置き換える。さらに、色成分ごとに透過色 $t$ を設定してロスレス符号化する。

#### 【0138】

従って、原画像をブロックに分割して前景/背景判定を行うことで、前景画素の色を限定してインデックス値に置き換えられる可能性が高くなり、ロスレス符号化の効率が向上する。つまり、実施例1などの効果に加えて、さらに、文字領域と写真領域が混在する原画像を効率的に圧縮符号化することができる。

#### 【0139】

##### 【他の実施例】

上記の実施例では、例えば、以下の各手段を備える画像処理装置を説明した。前景画素と背景画素からなる画像データを入力する入力手段。画像データにおいて、前景画素として出現する画素値のヒストグラムに基づいて、背景画素の画素値に前景画素として出現しない画素値を設定することにより、第1の画像データを生成する第1の生成手段。第1の画像データを符号化する第1の符号化手段。画像データにおいて、前景画素の画素値に背景画素の画素値に基づいた値を設定することにより、第2の画像データを生成する第2の生成手段。第2の画像データを符号化する第2の符号化手段である。

#### 【0140】

なお、上記実施例では、領域の分離方法の例をいくつか挙げたが、PDL画像の場合には、画像データに領域情報が付与された状態で入力されることがある。そのため、付与されている領域情報を用いて前景画像と背景画像の分離を行っても構わない。

#### 【0141】

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

#### 【0142】

また、本発明の目的は、上記実施例の機能を実現するコンピュータプログラムを記録した記憶媒体をシステムまたは装置に供給し、そのシステムまたは装置のコンピュータ（CPUやMPU）が前記コンピュータプログラムを実行することでも達成される。この場合、記憶媒体から読み出されたソフトウェア自体が上記実施例の機能を実現することになり、そのコンピュータプログラムと、そのコンピュータプログラムを記憶する、コンピュータが読み取り可能な記憶媒体は本発明を構成する。

10

20

30

40

50

## 【0143】

また、前記コンピュータプログラムの実行により上記機能が実現されるだけではない。つまり、そのコンピュータプログラムの指示により、コンピュータ上で稼働するオペレーティングシステム(OS)および/または第一の、第二の、第三の、…プログラムなどが実際の処理の一部または全部を行い、それによって上記機能が実現される場合も含む。

## 【0144】

また、前記コンピュータプログラムがコンピュータに接続された機能拡張カードやユニットなどのデバイスのメモリに書き込まれていてもよい。つまり、そのコンピュータプログラムの指示により、第一の、第二の、第三の、…デバイスのCPUなどが実際の処理の一部または全部を行い、それによって上記機能が実現される場合も含む。

10

## 【0145】

本発明を前記記憶媒体に適用する場合、その記憶媒体には、先に説明したフローチャートに対応または関連するコンピュータプログラムが格納される。

## 【図面の簡単な説明】

## 【0146】

【図1】実施例の符号化装置の構成例を示すブロック図、

【図2】周辺八画素を示す図、

【図3】符号データの構成例を示す図、

【図4】ロスレス符号化部の構成例を示すブロック図、

【図5】ハフマンテーブルの一例を示す図、

20

【図6】実施例の復号装置の構成例を示すブロック図、

【図7】画像合成部の処理を示すフローチャート、

【図8】コンピュータ装置の構成例を示すブロック図、

【図9】実施例2の領域判定部の処理を示すフローチャート、

【図10】注目画素の値と同一の値をもつ画素が周辺八画素に存在しない場合を説明する図、

【図11】周辺24画素を示す図、

【図12】実施例3の符号化装置の構成例を示すブロック図、

【図13】前景画素変換部の構成例を示すブロック図、

【図14】有効レベル数カウンタが有効レベル数をカウントするための二値配列を示す図

30

、

【図15】有効レベル数カウンタの処理を示すフローチャート、

【図16】インデックス変換用のテーブル例を示す図、

【図17】インデックステーブル生成部の処理を示すフローチャート、

【図18】タイル単位の符号データの構成例を示す図である。

【図19A】前景画素のヒストグラム例を示す図、

【図19B】前景画素のヒストグラム例を示す図、

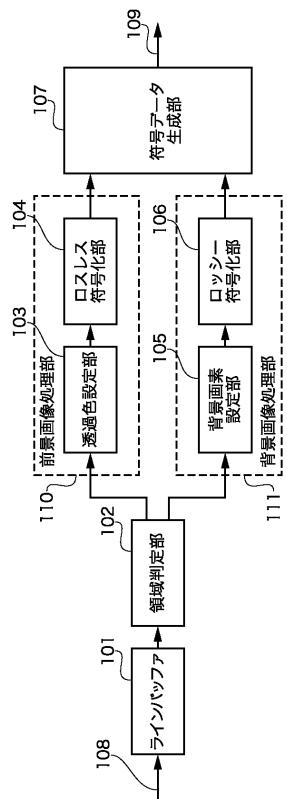
【図19C】前景画素のヒストグラム例を示す図、

【図19D】前景画素のヒストグラム例を示す図、

【図20】輝度値が異なる前景画素が混在する画像例を示す図である。

40

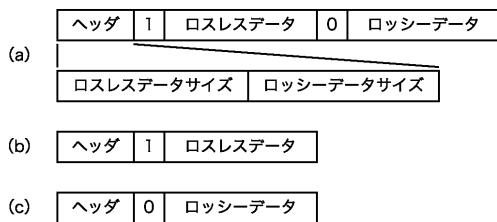
【図1】



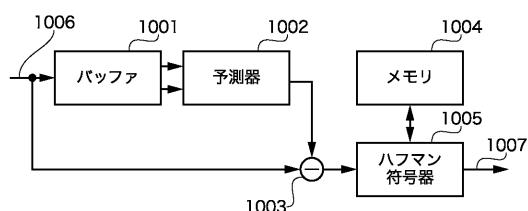
【図2】

a	b	c
d	X	e
f	g	h

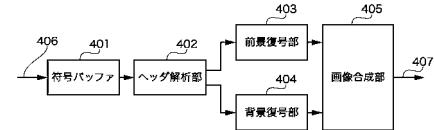
【図3】



【図4】



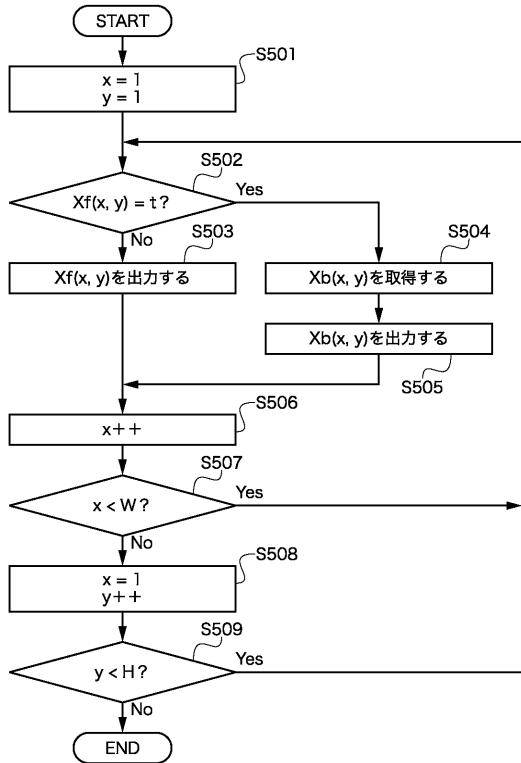
【図6】



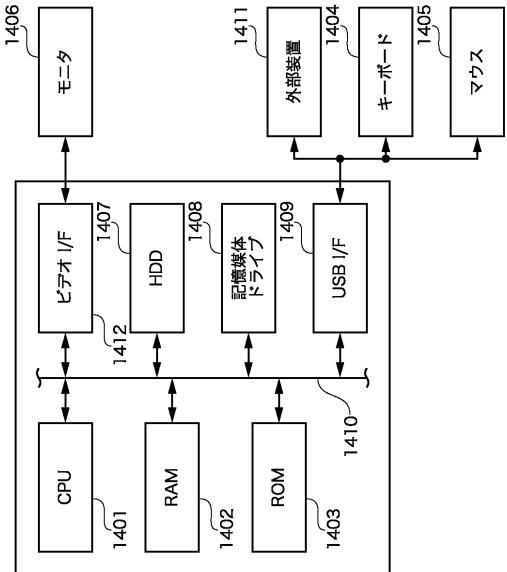
【図5】

シンボル	符号
255	1 1 1 ..... 1 1 1 0
254	1 1 1 ..... 1 1 0 0
...	
3	1 1 1 0 0
2	1 1 0 0
1	1 0 0
0	0 0
-1	0 1
-2	1 0 1
-3	1 1 0 1
...	
-254	1 1 1 ..... 1 1 0 1
-255	1 1 1 ..... 1 1 1 1

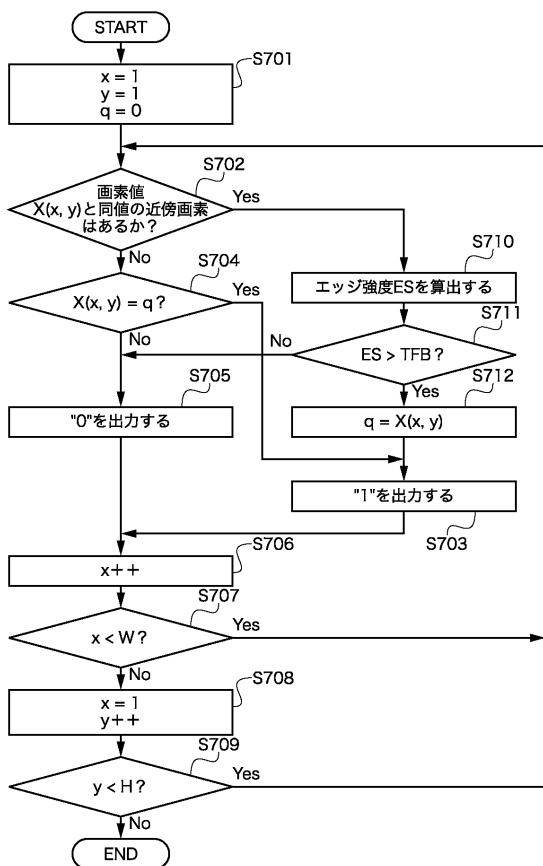
【図7】



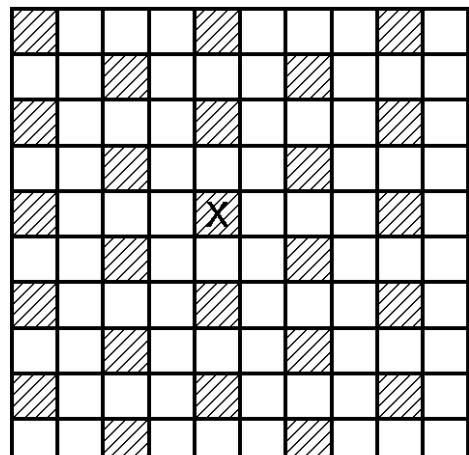
【図8】



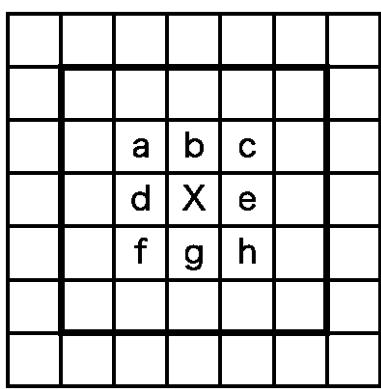
【図9】



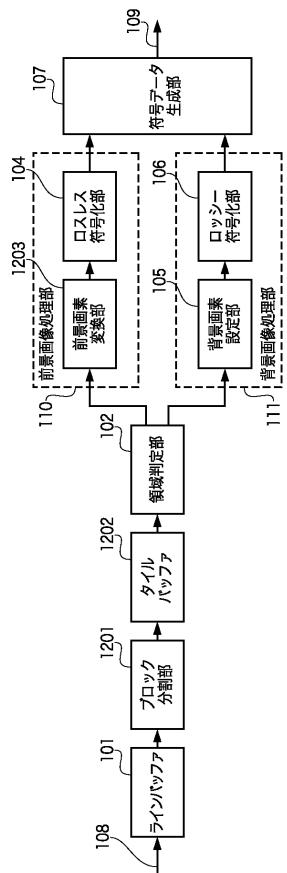
【図10】



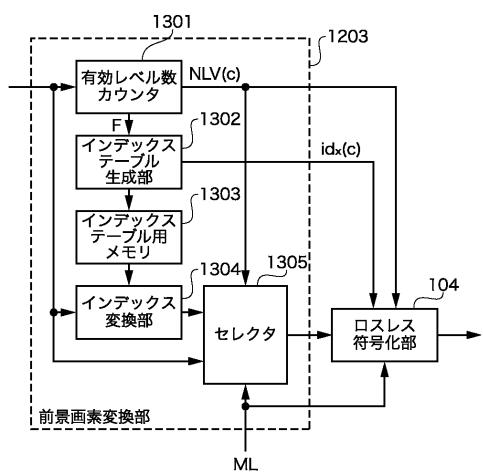
【図11】



【図12】



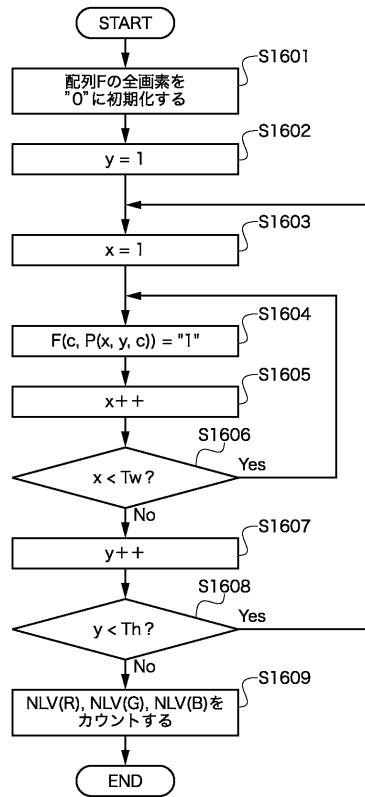
【図13】



【図14】

輝度値 (i)	$F(R, i)$	$F(G, i)$	$F(B, i)$
0			
1			
2			
3			
...			
254			
255			

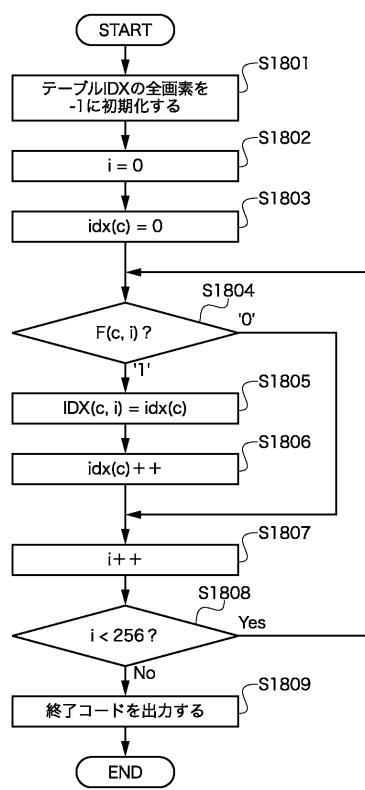
【図15】



【図16】

値	IDX(R, i)	IDX(G, i)	IDX(B, i)
0	0		
1			
2	1		
3	2		

【図17】

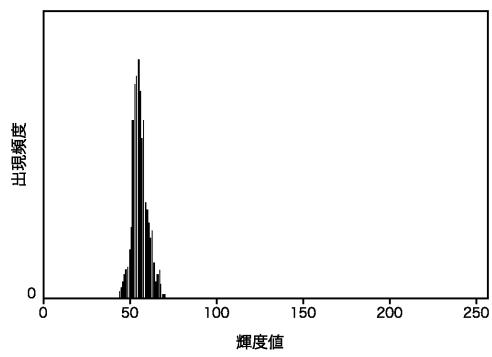


【図18】

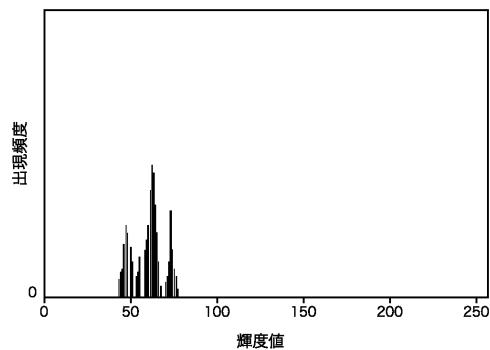
1バイト	変換テーブル情報	R符号化データ	G符号化データ	B符号化データ
7.....3	2	1	0	
0.....0	0: NLV(R) ≥ ML 1: NLV(R) < ML	0: NLV(G) ≥ ML 1: NLV(G) < ML	0: NLV(B) ≥ ML 1: NLV(B) < ML	

(b) 0 R符号化データ G符号化データ B符号化データ

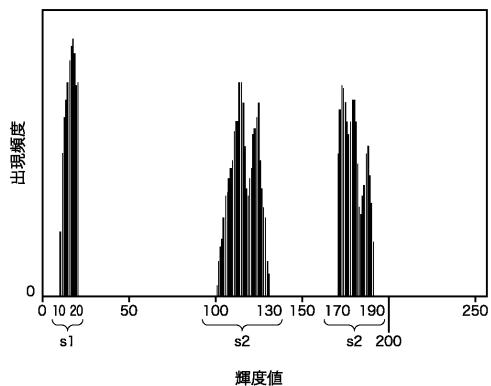
【図19 A】



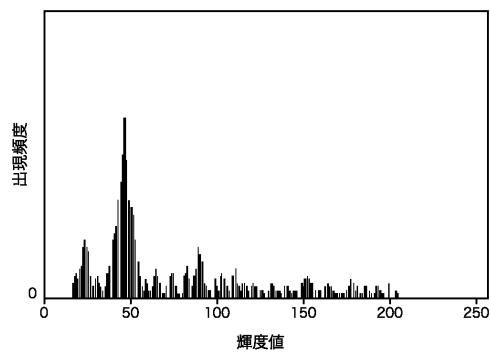
【図 1 9 B】



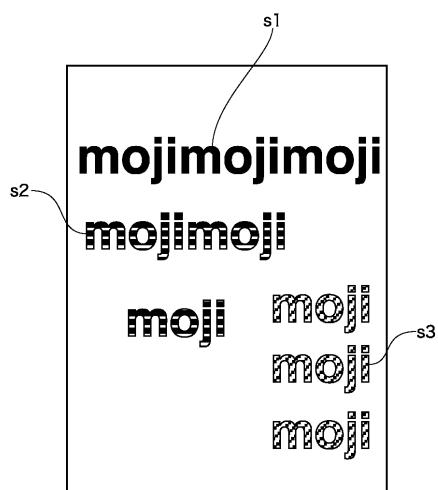
【図 1 9 D】



【図 1 9 C】



【図 2 0】



---

フロントページの続き

(72)発明者 松本 友希

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 堀井 啓明

(56)参考文献 特開2007-089136 (JP, A)

特開2005-151382 (JP, A)

特開平9-289586 (JP, A)

(58)調査した分野(Int.Cl., DB名)

H04N1/41-1/419

H04N7/12-7/137