

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5173873号  
(P5173873)

(45) 発行日 平成25年4月3日(2013.4.3)

(24) 登録日 平成25年1月11日(2013.1.11)

(51) Int.Cl. F I  
**HO4N 1/41 (2006.01)** HO4N 1/41 B  
**HO4N 7/26 (2006.01)** HO4N 7/13 Z

請求項の数 20 (全 39 頁)

(21) 出願番号	特願2009-20743 (P2009-20743)	(73) 特許権者	000001007
(22) 出願日	平成21年1月30日 (2009.1.30)		キヤノン株式会社
(65) 公開番号	特開2010-154486 (P2010-154486A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成22年7月8日 (2010.7.8)	(74) 代理人	100076428
審査請求日	平成24年1月17日 (2012.1.17)		弁理士 大塚 康徳
(31) 優先権主張番号	特願2008-297098 (P2008-297098)	(74) 代理人	100112508
(32) 優先日	平成20年11月20日 (2008.11.20)		弁理士 高柳 司郎
(33) 優先権主張国	日本国 (JP)	(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】 画像符号化装置及びその制御方法

(57) 【特許請求の範囲】

【請求項1】

画像データを圧縮符号化する画像符号化装置であって、  
 符号化対象のオリジナル画像データから2×2画素で構成されるブロックを入力する入力手段と、

入力したブロック内の、予め設定された位置の1つの画素データをサンプリングすることで、前記オリジナル画像データよりも小さいサイズの縮小画像データを生成し、出力する縮小画像生成手段と、

前記縮小画像生成手段で生成された前記縮小画像データから前記オリジナル画像データを復元するため、各ブロック中の非サンプリング対象となった3つの画素データを復元するための符号化データを生成する符号化手段とを備え、

前記符号化手段は、

着目ブロック中の前記サンプリング対象の画素をX、前記着目ブロック中の非サンプリング対象の画素をXa、Xb、Xcと表わしたとき、

条件(a)：画素Xa、Xb、Xcの全てが、前記着目ブロック中の前記画素Xから復元できるか、

条件(b)：前記画素Xa、Xb、Xcの全てが、前記着目ブロックに隣接する3つのブロックのサンプリング対象となる画素から復元できるか、

を判定する判定手段と、

該判定手段によって前記着目ブロックが前記条件(a)を満たすと判定された場合、第

1の付加情報を前記着目ブロックの符号化データとして生成し、

前記判定手段によって、前記着目ブロックが前記条件(a)を満たさず、前記条件(b)を満たすと判定された場合、前記第1の付加情報とは異なる第2の付加情報を前記着目ブロックの符号化データとして生成し、

前記判定手段によって、前記着目ブロックが前記条件(a)、(b)のいずれも満たさないと判定された場合、前記第1、第2の付加情報とも異なる第3の付加情報を前記着目ブロックの符号化データとして生成する付加情報生成手段と

を備えることを特徴とする画像符号化装置。

【請求項2】

前記第1の付加情報は1ビット、前記第2の付加情報は前記第1の付加情報とは異なる値のビットで始まる2ビット、前記第3の付加情報は前記第2の付加情報の1ビット目が同じで2ビット目が異なる2ビット及び前記画素 $X_a$ 、 $X_b$ 、 $X_c$ の3画素分のデータで構成されることを特徴とする請求項1に記載の画像符号化装置。

10

【請求項3】

更に、前記付加情報生成手段で生成される水平方向にM個、垂直方向にN個(M、Nは2以上の整数)のブロックの符号化データを入力する符号化データ入力手段と、

該符号化データ入力手段で入力した $M \times N$ 個のブロックの符号化データを、当該 $M \times N$ 個のブロックで構成される1つのタイルの符号化データとして統合するタイル統合手段とを備え、

前記タイル統合手段は、

20

前記 $M \times N$ 個のブロックの全符号化データが、前記第1乃至第3の付加情報のいずれか1種類のみで構成される場合には、タイルに統合したことを示すフラグ情報を出力し、当該フラグ情報に後続して符号化データを出力する第1の統合手段と、

前記 $M \times N$ 個のブロックの全符号化データの中に、少なくとも2種類の符号化データが存在する場合、タイルへの非統合を示すフラグ情報を出力し、着目タイル内の水平方向に並ぶM個のブロックの符号化データで構成されるブロックラインを単位に、各符号化データを統合するブロックライン統合手段とを備え、

前記ブロックライン統合手段は、

着目ブロックラインのM個の符号化データが、前記第1乃至第3の付加情報のいずれか1種類のみで構成される場合には、ブロックラインに統合したことを示すフラグ情報を出力するとともに、当該フラグ情報に後続して前記符号化データを出力する第2の統合手段と、

30

前記着目ブロックラインのM個の符号化データの中に、少なくとも2種類の符号化データが存在する場合、ブロックラインへの非統合を示すフラグ情報を出力するとともに、着目ブロックラインに含まれるM個のブロックの符号化データを出力する手段とを備えることを特徴とする請求項1又は2に記載の画像符号化装置。

【請求項4】

前記符号化手段は、

$X = X_a = X_b = X_c$ を満たす場合、前記条件(a)を満たすと判定し、

画素 $X_a$ を復元するために参照する隣接ブロック内の前記画素 $X_a$ に隣接するサンプリング対象となる画素を $X_1$ 、画素 $X_b$ を復元するために参照する隣接ブロック内の前記画素 $X_b$ に隣接するサンプリング対象となる画素を $X_2$ 、画素 $X_c$ を復元するために参照する隣接ブロック内の前記画素 $X_c$ に隣接するサンプリング対象となる画素を $X_3$ と定義したとき、

40

$X_a = X_1$  且つ、

$X_b = X_2$  且つ、

$X_c = X_3$

を満たす場合、前記条件(b)を満たすと判定することを特徴とする請求項1乃至3のいずれか1項に記載の画像符号化装置。

【請求項5】

50

更に、前記付加情報生成手段で得られた符号化データのデータ量と、予め設定された目標符号量とを比較する第1の比較手段と、

前記第1の比較手段による比較結果、前記符号化データのデータ量が前記目標符号量を超える場合、前記入力手段による入力されたオリジナル画像データにおける各ブロックの画素値の値を修整することで、ブロックが表わす色数を減色する減色手段とを備え、

前記符号化手段は、前記減色手段で減色後のブロックについて再符号化を実行することを特徴とする請求項1乃至4のいずれか1項に記載の画像符号化装置。

【請求項6】

更に、前記付加情報生成手段が生成した前記第3の付加情報の個数と、前記オリジナル画像データに含まれる総ブロック数で決定される閾値とを比較する第2の比較手段と、

該第2の比較手段の比較結果に基づき、前記減色手段による減色の度合を設定するパラメータを設定する設定手段とを備えることを特徴とする請求項5に記載の画像符号化装置。

【請求項7】

更に、前記付加情報生成手段で得られた符号化データのデータ量と、予め設定された目標符号量とを比較する第1の比較手段と、

該第1の比較手段による比較結果が、前記符号化データのデータ量が前記目標符号量を超えることを示す場合、前記付加情報生成手段が生成した前記第3の付加情報の個数と、前記オリジナル画像データに含まれる総ブロック数で決定される閾値とを比較する第2の比較手段と、

該第2の比較手段による比較結果が、前記第3の付加情報の個数が前記閾値を超えることを示す場合、前記オリジナル画像データを、当該オリジナル画像データが持つ解像度よりも低い解像度に変換し、符号化データとして出力する解像度変換手段と、

該第2の比較手段による比較結果が、前記第3の付加情報の個数が前記閾値以下であるところを示す場合、前記入力手段による入力されたオリジナル画像データにおける各ブロックの画素値の値を修整することで、ブロックが表わす色数を減色し、減色後のブロックを前記符号化手段に出力することで再符号化を実行する減色手段と

を備えることを特徴とする請求項1乃至4のいずれか1項に記載の画像符号化装置。

【請求項8】

画像データを圧縮符号化する画像符号化装置であって、

符号化対象のオリジナル画像データから2×2画素で構成されるブロック内の、予め設定された位置の1つの画素データをサンプリングすることで、前記オリジナル画像データよりも小さいサイズの縮小画像データを生成し、出力する縮小画像生成手段と、

オリジナル画像データの中の水平方向に2M画素、垂直方向に2N画素（M、Nは2以上の整数）の画素データで構成されるタイルを入力する入力手段と、

入力したタイル中のM×N個のブロックそれぞれについて、前記縮小画像データを生成する際のサンプリング対象の画素を除く非サンプリング対象の3つ画素データを復元するための符号化データを、前記タイル単位に生成するタイル符号化手段とを備え、

前記タイル符号化手段は、

1つのブロック内の非サンプリング対象となった3画素の全てが当該ブロック内のサンプリング対象の画素から復元可能なブロックをフラットブロックと定義し、1つのブロック内の非サンプリング対象となった3画素の少なくとも1つが、当該ブロック内のサンプリング対象の画素から復元できないブロックをノンフラットブロックと定義したとき、

前記着目タイル内の全ブロックが前記フラットブロックであるか、少なくとも1つがノンフラットブロックであるかを判定し、当該判定結果を示すフラグ情報を出力する第1の判定手段と、

該第1の判定手段によって、前記着目タイル内の少なくとも1つが前記ノンフラットブロックであると判定した場合、前記着目タイル内の水平方向に並ぶM個のブロックで構成されるブロックラインを単位に、符号化するブロックライン符号化手段とを備えることを特徴とする画像符号化装置。

10

20

30

40

50

## 【請求項 9】

前記ブロックライン符号化手段は、

1 タイルの符号化の開始時に、初期値として全ブロックが前記ノンフラットブロックであることを示す参照ノンフラットブロックライン情報を生成する生成手段と、

前記着目ブロックライン内の全ブロックが前記フラットブロックであるか、少なくとも1つがノンフラットブロックであるかを判定し、当該判定結果を示すフラグ情報を出力する第2の判定手段と、

該第2の判定手段によって、前記着目ブロックライン内の少なくとも1つが前記ノンフラットブロックであると判定した場合、着目ブロックラインを構成する各ブロックのフラットブロック、ノンフラットブロックの種類が、前記参照ノンフラットブロックライン情報と同じであるか否かを判定し、当該判定結果を示すフラグ情報を出力する第3の判定手段と、

該第3の判定手段によって、前記着目ブロックラインの各ブロックのフラットブロック、ノンフラットブロックの種類が、前記参照ノンフラットブロックライン情報と異なると判定した場合、前記着目ブロックラインを構成する各ブロックの種類を示す情報を出力するとともに、前記参照ノンフラットブロックライン情報を、前記着目ブロックラインを構成する各ブロックの種類を示す情報で更新する手段と、

前記第2の判定手段、前記第3の判定手段から出力される情報に基づき、着目タイル内のノンフラットブロックを特定し、当該特定されたノンフラットブロックの中の非サンプリング対象の3画素を符号化するノンフラットブロック符号化手段と

を備えることを特徴とする請求項8に記載の画像符号化装置。

## 【請求項 10】

前記ノンフラットブロック符号化手段は、

前記ノンフラットブロックであると判定された着目ブロック内の非サンプリングの3画素が、前記着目ブロックに隣接する3つのブロックのサンプリング対象となる画素から復元できるか否かを判定し、当該判定結果を示すフラグ情報を出力する第4の判定手段と

、  
該第4の判定手段により、前記着目ブロック内の非サンプリングの3画素の少なくとも1つが、前記着目ブロックに隣接するブロックのサンプリング対象となる画素から復元できないと判定した場合、前記着目ブロックを構成する4画素が表わす色数が2であるか、それより多いかを判定し、当該判定結果を示すフラグ情報を出力する第5の判定手段と

、  
該第5の判定手段により、前記着目ブロックを構成する4画素が表わす色数が2より大きいと判定した場合、前記着目ブロック内の非サンプリングの3つの画素データを出力する第1の出力手段と、

前記第5の判定手段により、前記着目ブロックを構成する4画素が表わす色数が2色であると判定した場合、予め設定された2×2画素の複数の2色のパターンのいずれに該当するかを示す情報を出力する第2の出力手段と、

該第2の出力手段による前記情報の出力に後続して、前記着目ブロック内の前記非サンプリング対象の画素のうち、前記サンプリング対象となった画素の色とは異なる色の画素が、前記着目ブロックに隣接する3つのブロックのサンプリング対象の画素のいずれかと等しいか否かを判定する第6の判定手段と、

該第6の判定手段により等しい画素が存在すると判定した場合、当該等しいと判定された画素の位置を特定する情報を出力する第3の出力手段と、

前記第6の判定手段により等しい画素が存在しないと判定した場合、前記第6の判定手段が出力する前記情報とは異なる情報を出力し、且つ、前記着目ブロック内の前記非サンプリング対象の画素のうち、前記サンプリング対象となった画素の色とは異なる色の画素の画素データを出力する第4の出力手段と

を備えることを特徴とする請求項9に記載の画像符号化装置。

## 【請求項 11】

10

20

30

40

50

前記第 4 の判定手段は、

前記着目ブロック内のサンプリング対象の画素を  $X$ 、前記着目ブロック内の非サンプリング対象の 3 つの画素を  $X_a$ 、 $X_b$ 、 $X_c$ 、前記着目ブロックに隣接するブロック中の、前記画素  $X_a$  に隣接するサンプリング対象となる画素を  $X_1$ 、前記着目ブロックに隣接するブロック中の、前記画素  $X_b$  に隣接するサンプリング対象となる画素を  $X_2$ 、前記着目ブロックに隣接するブロック中の、前記画素  $X_c$  に隣接するサンプリング対象となる画素を  $X_3$  と定義したとき、予め設定された閾値  $T_{h1}$  について

$$\begin{aligned} &|X_a - X_1| < T_{h1} \quad \text{且つ、} \\ &|X_b - X_2| < T_{h1} \quad \text{且つ、} \\ &|X_c - X_3| < T_{h1} \end{aligned}$$

10

を満たす場合、着目ブロック内の非サンプリングの 3 画素が、前記着目ブロックに隣接する 3 つのブロックのサンプリング対象となる画素から復元できると判定することを特徴とする請求項 10 に記載の画像符号化装置。

【請求項 12】

前記着目ブロック内のサンプリング対象の画素を  $X$ 、前記着目ブロック内の非サンプリング対象の 3 つの画素を  $X_a$ 、 $X_b$ 、 $X_c$ 、前記着目ブロックに隣接するブロック中の、前記画素  $X_a$  に隣接するサンプリング対象となる画素を  $X_1$ 、前記着目ブロックに隣接するブロック中の、前記画素  $X_b$  に隣接するサンプリング対象となる画素を  $X_2$ 、前記着目ブロックに隣接するブロック中の、前記画素  $X_c$  に隣接するサンプリング対象となる画素を  $X_3$ 、予め設定された正の閾値を  $T_{h2}$  と定義したとき、

20

以下の条件 (b) を満たすか否かを判定する第 7 の判定手段と、

条件 (b) :

$$\begin{aligned} &|X_a - (X + X_1) / 2| < T_{h2}、\text{且つ、} \\ &|X_b - (X + X_2) / 2| < T_{h2}、\text{且つ、} \\ &|X_c - (X + X_3) / 2| < T_{h2} \end{aligned}$$

前記第 5 の判定手段で、前記着目ブロックを構成する 4 画素が表わす色数が 2 より多いと判定した場合、前記第 7 の判定手段の判定結果を示すフラグ情報を出力する第 5 の出力手段とを備え、

前記第 5 の出力手段が出力するフラグ情報が、前記条件 (b) を満たさないことを示す場合、前記第 1 の出力手段が、前記着目ブロック内の非サンプリングの 3 つの画素データを出力することを特徴とする請求項 10 又は 11 に記載の画像符号化装置。

30

【請求項 13】

前記タイル符号化手段により生成された符号化データの一部を切り捨てることにより、符号量を制御することを特徴とする請求項 8 に記載の画像符号化装置。

【請求項 14】

更に、前記タイル符号化手段で得られた符号化データのデータ量と、予め設定された目標符号量とを比較する第 1 の比較手段と、

前記第 1 の比較手段による比較結果、前記符号化データのデータ量が前記目標符号量を超える場合、前記入力手段による入力されたオリジナル画像データにおける各ブロックの画素値の値を修整することで、ブロックが表わす色数を減色する減色手段とを備え、

40

前記タイル符号化手段は、前記減色手段で減色後のブロックについて再符号化を実行することを特徴とする請求項 10 に記載の画像符号化装置。

【請求項 15】

更に、前記第 1 の出力手段、前記第 4 の出力手段が画素データを出力した回数と、前記オリジナル画像データに含まれる総ブロック数で決定される閾値とを比較する第 2 の比較手段と、

該第 2 の比較手段の比較結果に基づき、前記減色手段による減色の度合を設定するパラメータを設定する設定手段とを備えることを特徴とする請求項 14 に記載の画像符号化装置。

【請求項 16】

50

更に、前記タイル符号化手段で得られた符号化データのデータ量と、予め設定された目標符号量とを比較する第1の比較手段と、

該第1の比較手段による比較結果が、前記符号化データのデータ量が前記目標符号量を超えることを示す場合、前記第1の出力手段、前記第4の出力手段が画素データを出力した回数と、前記オリジナル画像データに含まれる総ブロック数で決定される閾値とを比較する第2の比較手段と、

該第2の比較手段による比較結果が、前記回数が前記閾値を超えることを示す場合、前記オリジナル画像データを、当該オリジナル画像データが持つ解像度よりも低い解像度に変換し、符号化データとして出力する解像度変換手段と、

該第2の比較手段による比較結果が、前記回数が前記閾値以下であることを示す場合、前記入力手段による入力されたオリジナル画像データにおける各ブロックの画素値の値を修整することで、ブロックが表わす色数を減色し、減色後のブロックを前記タイル符号化手段に出力することで再符号化を実行する減色手段と

を備えることを特徴とする請求項10に記載の画像符号化装置。

【請求項17】

画像データを圧縮符号化する画像符号化装置の制御方法であって、

符号化対象のオリジナル画像データから2×2画素で構成されるブロックを入力する入力工程と、

入力したブロック内の、予め設定された位置の1つの画素データをサンプリングすることで、前記オリジナル画像データよりも小さいサイズの縮小画像データを生成し、出力する縮小画像生成工程と、

前記縮小画像生成工程で生成された前記縮小画像データから前記オリジナル画像データを復元するため、各ブロック中の非サンプリング対象となった3つの画素データを復元するための符号化データを生成する符号化工程とを有し、

前記符号化工程は、

着目ブロック中の前記サンプリング対象の画素を $X$ 、前記着目ブロック中の非サンプリング対象の画素を $X_a$ 、 $X_b$ 、 $X_c$ と表わしたとき、

条件(a)：画素 $X_a$ 、 $X_b$ 、 $X_c$ の全てが、前記着目ブロック中の前記画素 $X$ から復元できるか、

条件(b)：前記画素 $X_a$ 、 $X_b$ 、 $X_c$ の全てが、前記着目ブロックに隣接する3つのブロックのサンプリング対象となる画素から復元できるか、

を判定する判定工程と、

該判定工程によって前記着目ブロックが前記条件(a)を満たすと判定された場合、第1の付加情報を前記着目ブロックの符号化データとして生成し、

前記判定工程によって、前記着目ブロックが前記条件(a)を満たさず、前記条件(b)を満たすと判定された場合、前記第1の付加情報とは異なる第2の付加情報を前記着目ブロックの符号化データとして生成し、

前記判定工程によって、前記着目ブロックが前記条件(a)、(b)のいずれも満たさないと判定された場合、前記第1、第2の付加情報とも異なる第3の付加情報を前記着目ブロックの符号化データとして生成する付加情報生成工程と

を有することを特徴とする画像符号化装置の制御方法。

【請求項18】

画像データを圧縮符号化する画像符号化装置の制御方法であって、

符号化対象のオリジナル画像データから2×2画素で構成されるブロック内の、予め設定された位置の1つの画素データをサンプリングすることで、前記オリジナル画像データよりも小さいサイズの縮小画像データを生成し、出力する縮小画像生成工程と、

オリジナル画像データの中の水平方向に2M画素、垂直方向に2N画素(M、Nは2以上の整数)の画素データで構成されるタイルを入力する入力工程と、

入力したタイル中のM×N個のブロックそれぞれについて、前記縮小画像データを生成する際のサンプリング対象の画素を除く非サンプリング対象の3つ画素データを復元する

10

20

30

40

50

ための符号化データを、前記タイル単位に生成するタイル符号化工程とを有し、  
前記タイル符号化工程は、

1つのブロック内の非サンプリング対象となった3画素の全てが当該ブロック内のサンプリング対象の画素から復元可能なブロックをフラットブロックと定義し、1つのブロック内の非サンプリング対象となった3画素の少なくとも1つが、当該ブロック内のサンプリング対象の画素から復元できないブロックをノンフラットブロックと定義したとき、

前記着目タイル内の全ブロックが前記フラットブロックであるか、少なくとも1つがノンフラットブロックであるかを判定し、当該判定結果を示すフラグ情報を出力する第1の判定工程と、

該第1の判定工程によって、前記着目タイル内の少なくとも1つが前記ノンフラットブロックであると判定した場合、前記着目タイル内の水平方向に並ぶM個のブロックで構成されるブロックラインを単位に、符号化するブロックライン符号化工程と有する

ことを特徴とする画像符号化装置の制御方法。

【請求項19】

コンピュータに読み込ませ実行させることで、前記コンピュータを請求項1乃至16のいずれか1項に記載の画像符号化装置として機能させることを特徴とするコンピュータプログラム。

【請求項20】

請求項19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コンピュータ可読記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は画像符号化技術に関するものである。

【背景技術】

【0002】

高解像度で広色域な画像データは、データ容量が膨大になり、メモリコストも膨大になり、画像処理やデータ転送などで時間がかかってしまうなど、データの扱いが困難になるという問題があった。そのため、従来、複数画素に対して、1つの色情報と、色の配置を表す情報を保持することで、オリジナル解像度の画像よりも、小さい容量のデータに変換する方法が提案されている（特許文献1）。

【0003】

また、画像データの中の、文字画像部分と、写真画像部分を分離して、それぞれの画像に対して、縮小画像を生成する。そして、その情報を保持しておき、オリジナル解像度の画像を復元する場合には、着目している縮小画像の画素とその周辺画素から、復元対象画素を予測して復元する方法も提案されている（特許文献2）。

【特許文献1】特開2001-313834号公報

【特許文献2】特開平10-173913号公報

【発明の開示】

【発明が解決しようとする課題】

【0004】

解像度変換を行って生成された縮小画像に加えて、補間データを生成するような方法の場合、補間データには、少なくともオリジナル解像度の画像を再現するための、色情報と、その色を配置するための、配置情報とが必要となる。そのため、オリジナル解像度の画像の1画素に対して、少なくとも1つの配置情報を有することになり、補間データの符号量の削減効果は、 $1 / (1 \text{画素分のデータ量})$ が限界である。

【0005】

また、補間データを持たず、決まった法則により、復元対象画像の補間を行う方式においては、補間するデータは、推測された値にすぎず、オリジナル解像度の画像を可逆で再現できるわけではない。

10

20

30

40

50

## 【 0 0 0 6 】

本願発明は、上述の問題点に鑑みてなされたものである。そして、オリジナル画像データから縮小画像データを生成し、且つ、その縮小画像データに効率の良い符号化データを付加させることで、その縮小画像データからオリジナル画像データを復元することができる技術を提供しようとするものである。さらに、補間データを制御することで、可逆と非可逆の変換が容易に可能な技術を提供しようとするものである。

## 【課題を解決するための手段】

## 【 0 0 0 7 】

かかる課題を解決するため、例えば本発明の画像符号化装置は以下の構成を備える。すなわち、

画像データを圧縮符号化する画像符号化装置であって、

符号化対象のオリジナル画像データから  $2 \times 2$  画素で構成されるブロックを入力する入力手段と、

入力したブロック内の、予め設定された位置の1つの画素データをサンプリングすることで、前記オリジナル画像データよりも小さいサイズの縮小画像データを生成し、出力する縮小画像生成手段と、

前記縮小画像生成手段で生成された前記縮小画像データから前記オリジナル画像データを復元するため、各ブロック中の非サンプリング対象となった3つの画素データを復元するための符号化データを生成する符号化手段とを備え、

前記符号化手段は、

着目ブロック中の前記サンプリング対象の画素を  $X$ 、前記着目ブロック中の非サンプリング対象の画素を  $X_a$ 、 $X_b$ 、 $X_c$  と表わしたとき、

条件 ( a ) : 画素  $X_a$ 、 $X_b$ 、 $X_c$  の全てが、前記着目ブロック中の前記画素  $X$  から復元できるか、

条件 ( b ) : 前記画素  $X_a$ 、 $X_b$ 、 $X_c$  の全てが、前記着目ブロックに隣接する3つのブロックのサンプリング対象となる画素から復元できるか、

を判定する判定手段と、

該判定手段によって前記着目ブロックが前記条件 ( a ) を満たすと判定された場合、第1の付加情報を前記着目ブロックの符号化データとして生成し、

前記判定手段によって、前記着目ブロックが前記条件 ( a ) を満たさず、前記条件 ( b ) を満たすと判定された場合、前記第1の付加情報とは異なる第2の付加情報を前記着目ブロックの符号化データとして生成し、

前記判定手段によって、前記着目ブロックが前記条件 ( a )、( b ) のいずれも満たさないと判定された場合、前記第1、第2の付加情報とも異なる第3の付加情報を前記着目ブロックの符号化データとして生成する付加情報生成手段とを備える。

## 【発明の効果】

## 【 0 0 0 8 】

本発明によれば、縮小画像データからオリジナル画像データを復元するための効率の良い符号化データを生成することができる。

## 【発明を実施するための最良の形態】

## 【 0 0 0 9 】

以下添付図面に従って本発明に係る実施形態を詳細に説明する。

## 【 0 0 1 0 】

[ 第1の実施形態 ]

図2は、実施形態における画像処理装置のブロック図である。以下の説明から明らかになるが、本画像処理装置は、画像圧縮符号化にかかる構成と、復号にかかる構成が一体となっている例を説明するが、これらは別々の装置であっても構わない。

## 【 0 0 1 1 】

本実施形態の画像処理装置は、符号化対象となる画像データを外部から入力し、解像度変換を行うとともに、縮小画像からオリジナル解像度の画像を復元するための、付加情報

10

20

30

40

50



を生成するものである。なお、画像データの入力源は、画像データをファイルとして格納している記憶媒体とするが、イメージスキャナ等であっても良く、その種類は問わない。

【0012】

また、実施形態における符号化対象の画像データは輝度成分のみのモノクロ多値画像データとし、輝度成分は8ビット(0乃至255の256階調)として説明する。但し、これは実施形態の説明を簡単なものとするためであって、複数成分(例えば、RGBやCMYK)でも構わず、その色空間の種類は問わない。また、1成分のビット数も8ビットに限らず、8ビットを超えるビット数でも構わない。更に、符号化対象の画像は水平方向に $2 \times W$ 画素、垂直方向 $2 \times H$ 画素(共に偶数)により構成されるとして説明する。なお、実際の画像の水平、垂直方向のサイズは2の整数倍となるとは限らないが、その場合には、仮想的な固定値を有する画素が水平、垂直の末尾に存在するものとして処理すれば良いことを付言しておく。

10

【0013】

以下、図2における画像符号化装置における符号化処理を説明する。

【0014】

まず、画像入力部201から符号化対象となる画像データが順に入力される。本実施形態で処理対象とする画像データは、PDLレンダリング画像を対象とする。また、画素データの入力順序はラスタースキャン順とする。実施形態の場合、1画素=1コンポーネントであり、そのコンポーネントは8ビットであるものとしているので、画素値の取り得る範囲は0乃至255の範囲の非負の整数値である。画像の左上隅を座標(0,0)とし、水平右方向の画素位置 $x$ 、垂直下方向の画素位置 $y$ にある画素値を $P(x, y)$ と表す。例えば、位置 $(x, y) = (3, 4)$ にある画素が、128という輝度値を持つ場合、 $P(3, 4) = 128$ と表現する。また、以下の説明では、位置 $(x, y)$ の“画素”を表現する場合にも“ $P(x, y)$ ”という表現を用いることとする。

20

【0015】

解像度変換部202は、入力画像の $2 \times 2$ 個の画素データで構成されるブロックデータ内から1画素を抽出するサブサンプリングを行い、縮小画像を生成する。すなわち、この解像度変換部202はオリジナルの画像データから、それよりもサイズの小さい縮小画像を生成する縮小画像生成部として機能する。

【0016】

図3はオリジナル画像とその中の或るブロック $B_n$ に含まれる $2 \times 2$ 画素を示している。図示の如く、着目ブロック $B_n$ の中の画素を左上からラスタースキャン順に、 $B_n(0, 0)$ 、 $B_n(0, 1)$ 、 $B_n(1, 0)$ 、 $B_n(1, 1)$ と表す。

30

【0017】

着目ブロック $B_n$ の1つ前のブロックを $B_{n-1}$ 、着目ブロック $B_n$ の次のブロックを $B_{n+1}$ と表現することとする。本実施形態では、着目ブロック $B_n$ のうち、 $B_n(0, 0)$ の位置、すなわち、左上隅の位置の画素を抽出(サンプリング)し、縮小画像の1画素とする。画像中の全てのブロック $B_0 \sim B_{W \cdot H - 1}$ についてサンプリングを行い、水平方向 $2W$ 、垂直方向 $2H$ 画素で構成されるオリジナル画像データから、水平方向 $W$ 、垂直方向 $H$ の縮小画像を生成する。すなわち、縮小画像に含まれる画素数は、オリジナルの画像に含まれる画素数の $1/4$ となる。生成された縮小画像は、格納部204へ出力され、格納される。

40

【0018】

補間データ生成部203は、縮小画像からオリジナル画像を復元するために必要な情報を生成する。1つのブロック内の1画素が縮小画像の1画素としてサンプリングされたわけであるから、そのブロックを復元するためには残りの、非サンプリング対象であった3画素を復元できればよい。そこで、本実施形態の補間データ生成部203は、ブロック単位に、そのブロック内の残りの非サンプリング対象の3画素がどのような値を持つのか、または、どのようにすれば復元できるのかを示す情報を符号化データとして生成する。

【0019】

50

図4は1つのブロックに含まれる4つの画素を示している。ここで、図示の画素Xは縮小画像の画素にそのまま利用された画素を示し、画素X<sub>a</sub>、X<sub>b</sub>、X<sub>c</sub>が復元対象となる画素を示している。以下、着目ブロックB<sub>n</sub>のB<sub>n</sub>(0,0)の位置の画素をXと表現し、B<sub>n</sub>(0,1)、B<sub>n</sub>(1,0)、B<sub>n</sub>(1,1)の位置を、それぞれX<sub>a</sub>、X<sub>b</sub>、X<sub>c</sub>という表現を用いるものとする。

【0020】

図1は、本実施形態における補間データ生成部203の処理の流れを示したフローチャートである。以下、図1のフローチャートに従って、本実施形態の補間データ生成部203の処理フローを説明する。

【0021】

まず、ステップS101において、着目ブロックB<sub>n</sub>に含まれる2×2個の画素と、着目ブロックB<sub>n</sub>に隣接する3つのブロックからそれぞれ1画素を入力する。ここで、着目ブロックに隣接する3つのブロックとは、右隣に位置するブロックB<sub>n+1</sub>、直下に位置するブロックB<sub>n+W</sub>、斜め右下のブロックB<sub>n+W+1</sub>である。そして、それぞれのブロックから入力する画素はB<sub>n+1</sub>(0,0)、直下に位置するブロックB<sub>n+W</sub>(0,0)、斜め右下のブロックB<sub>n+W+1</sub>(0,0)である。すなわち、これら3つの画素は、縮小画像に利用される画素であることに注意されたい。

【0022】

ステップS102で、パラメータnを“0”で初期化する。このパラメータnは、ブロックを特定するためのものである。

【0023】

次いで、ステップS103では、着目ブロック内の4つの画素が互いに等しいか否かを判定する。すなわち、次式(1)を満足するか否かを判定する。

$$B_n(0,0) = B_n(0,1) = B_n(1,0) = B_n(1,1) \quad \dots (1)$$

【0024】

着目ブロックB<sub>n</sub>が、式(1)の成立するブロックである場合(YES)には、縮小画像中の画素Xの値は、復元対象の画素X<sub>a</sub>、X<sub>b</sub>、X<sub>c</sub>の値でもあることになる。そこで、このような場合には、処理をステップS105に遷移させる。

【0025】

一方、着目ブロックB<sub>n</sub>について式(1)が成り立たないと判定した場合(ステップS103でNOの場合)、すなわち、着目ブロック中の画素Xから残りの3画素X<sub>a</sub>、X<sub>b</sub>、X<sub>c</sub>を復元できないと判定した場合はステップS104へ遷移する。

【0026】

ステップS104では、着目ブロックに隣接する3つのブロックB<sub>n+1</sub>、B<sub>n+W</sub>、B<sub>n+W+1</sub>における画素B<sub>n+1</sub>(0,0)、B<sub>n+W</sub>(0,0)、B<sub>n+W+1</sub>(0,0)と、着目ブロック内の画素B<sub>n</sub>(0,1)、画素B<sub>n</sub>(1,0)、画素B<sub>n</sub>(1,1)とを比較し、次式(2)が満たすか否かを判定する。

$$B_n(0,1) = B_{n+1}(0,0)$$

且つ

$$B_n(1,0) = B_{n+W}(0,0)$$

且つ

$$B_n(\underline{1},1) = B_{n+W+1}(0,0) \quad \dots (2)$$

【0027】

先に説明したように、画素B<sub>n+1</sub>(0,0)、B<sub>n+W</sub>(0,0)、B<sub>n+W+1</sub>(0,0)は、縮小画像を生成するために、それぞれのブロックB<sub>n+1</sub>、B<sub>n+W</sub>、B<sub>n+W+1</sub>からサンプリングされた画素であることに注意されたい。

【0028】

式(2)に示すように、ペア{B<sub>n</sub>(0,1)、B<sub>n+1</sub>(0,0)}、{B<sub>n</sub>(1,0)、B<sub>n+W</sub>(0,0)}、及び、{B<sub>n</sub>(1,1)とB<sub>n+W+1</sub>(0,0)}との一致/不一致を判定するには、理由がある。

10

20

30

40

50

## 【 0 0 2 9 】

一般に、着目画素と、その着目画素の上下左右に位置する画素間の相関度は高いし、そのような画像が多い。そのため、着目画素の画素値を予測する場合には、その隣接する画素を予測のための参照画素として使用することが多い。しかし、ステップ S 1 0 4 の判定では、「着目ブロック内の 4 つの画素が互いに等しい」ブロックは処理対象としないため、この場合にはペア  $\{ B_n(0, 1), B_{n+1}(0, 0) \}$ 、 $\{ B_n(1, 0), B_{n+W}(0, 0) \}$ 、及び、 $\{ B_n(0, 1) \text{ と } B_{n+W+1}(0, 0) \}$  の相関が高くなる可能性が高いことを実験により得られた。

## 【 0 0 3 0 】

この理由で、 $B_n(0, 1)$  と  $B_{n+1}(0, 0)$  とが一致するか否かを判定するようにした。他のペア  $\{ B_n(1, 0), B_{n+W}(0, 0) \}$ 、及び、 $\{ B_n(0, 1) \text{ と } B_{n+W+1}(0, 0) \}$  を比較するのも同じ理由である。そして、先に示した式 ( 2 ) が満たされる場合 ( 3 つのペアが互いに等しい場合 )、短い符号語を割り当てることで情報量の削減できた。

10

## 【 0 0 3 1 】

さて、ステップ S 1 0 4 にて、式 ( 2 ) の成立すると判定した場合、着目ブロック  $B_n$  内の画素  $X_a$ 、 $X_b$ 、 $X_c$  は、縮小画像の画素からそのまま再現できることになる。なぜなら、画素  $X_a$  は、縮小画像の画素  $X_{n+1}$  ( $n = 0, 1, \dots, W \times H - 1$ ) を用いれば再現できる。画素  $X_b$  は、縮小画像の画素  $X_{n+W}$  を用いれば再現できる。そして、画素  $X_c$  は、縮小画像の画素  $X_{n+W+1}$  を用いれば再現できる。

20

## 【 0 0 3 2 】

但し、着目ブロック  $B_n$  が画像の右端に位置するとき、画素  $B_n(0, 1)$ 、 $B_n(1, 1)$  と比較する対象画素は存在しない。また、着目ブロック  $B_n$  が画像の下端に位置するとき、画素  $B_n(1, 0)$ 、 $B_n(1, 1)$  と比較する対象画素は存在しない。特に、着目ブロック  $B_n$  が画像の右下隅に位置する場合には、画素  $B_n(0, 1)$ 、 $B_n(1, 0)$ 、 $B_n(1, 1)$  と比較する対象画素が存在しない。このように、存在しない画素の値は適当な値、例えば “ 2 5 5 ” であるものとして比較する。なお、この値は、復号装置と共通な値になっていけばよいので、0 乃至 2 5 5 の範囲内であれば、どのような値であっても構わない。

## 【 0 0 3 3 】

以上のように、式 ( 2 ) の判定を行った結果、条件が成立する場合 ( Y E S ) には、ステップ S 1 0 6 へ処理を遷移する。

30

## 【 0 0 3 4 】

さて、着目ブロック  $B_n$  について式 ( 1 )、式 ( 2 ) のいずれも成立しない場合、着目ブロックの  $X_a$ 、 $X_b$ 、 $X_c$  を、縮小画像からは復元できないことになる。この場合、処理はステップ S 1 0 7 へ遷移する。

## 【 0 0 3 5 】

ステップ S 1 0 5 は、着目ブロックに対して、第 1 の付加情報を符号化データとして出力する。第 1 の付加情報は、オリジナル画像を復元する際に、縮小画像の着目画素から 3 つの画素を生成する、すなわち、「単純拡大する」という情報を示せば良い。そのため、図 5 に示すように、“ 1 ” の 1 ビットを第 1 の付加情報を出力する。

40

## 【 0 0 3 6 】

なお、復号装置側にて、実際に縮小画像からオリジナル画像まで拡大した画像を生成する場合、縮小画像中の着目画素を  $P(x, y)$  とするならば、拡大後の着目ブロック  $B_n$  の画素  $X$ 、 $X_a$ 、 $X_b$ 、 $X_c$  は次のようにして求めればよい。

$$X = X_a = X_b = X_c = P(x, y)$$

## 【 0 0 3 7 】

ステップ S 1 0 6 は、着目ブロックに対して、第 2 の付加情報を符号化データとして出力する。第 2 の付加情報は、オリジナル解像度を復元する際に、縮小画像の着目画素の隣接する 3 つの画素を利用することを示せばよい。この場合、図 5 に示すように、“ 0 1 ”

50

の2ビットのデータを第2の付加情報(符号化データ)として出力する。

【0038】

実際に縮小画像からオリジナル画像まで拡大した画像を生成する場合、縮小画像中の着目画素を $P(x, y)$ とするなら、拡大後の着目ブロック $B_n$ の画素 $X, X_a, X_b, X_c$ は次のようにして求めればよい。

$$X = B_n(0, 0) = P(x, y)$$

$$X_a = B_n(0, 1) = P(x + 1, y)$$

$$X_b = B_n(1, 0) = P(x, y + 1)$$

$$X_c = B_n(1, 1) = P(x + 1, y + 1)$$

【0039】

ステップS107に処理が進んだ場合、着目ブロック $B_n$ の画素 $X_a, X_b, X_c$ は縮小画像から再現できないことを意味する。そこで、本第1の実施形態では、第3の付加情報を出力する。第3の付加情報は、図5に示すように、上記第1、第2の付加情報と区別できるようにするため、先頭の2ビットを“00”とし、これに後続して $B_n(0, 1)$ 、 $B_n(1, 0)$ 、 $B_n(1, 1)$ の画素値の24ビットを出力する。

【0040】

なお、実施形態では、第1乃至第3の付加情報が図5に示すものとして説明したが、これは一例である。要するに、第1の付加情報を1ビットとしたとき、第2の付加情報は第1の付加情報とは異なる値のビットで始まる2ビットとすればよい。そして、第3の付加情報は第2の付加情報の1ビット目が同じで2ビット目が異なる2ビット+3画素分のデータで構成されるようにすればよい。

【0041】

以上の処理の流れにより、着目ブロックに対して、第1、第2、第3のいずれかの付加情報を出力する。

【0042】

ステップS108では、着目ブロックが最終ブロック(画像の右下隅のブロック)であるか否かを判断し、否の場合にはステップS109にて変数 $n$ を“1”だけ更新し、ステップS103に処理を戻す。また、ステップS108で、着目ブロックが最終ブロック(画像の右下隅のブロック)であると判断した場合には、本処理を終了する。

【0043】

以上の方法で、オリジナル解像度の画像を復元可能な付加情報を生成することができる。つまり、実施形態における補間データ生成部203を別な言い方をすれば上記の第1乃至第3の付加情報を生成する付加情報生成部ということができる。

【0044】

具体的な処理を、図6(a)、(b)に示す画像の例で説明する。

【0045】

図6(a)の画像の場合、まず、ブロック $B_0$ に着目すると、 $B_0(0, 0) = B_0(0, 1) = B_0(1, 0) = B_0(1, 1)$ が成立するため、第1の付加情報“1”を出力する。続くブロック $B_1, B_2, B_3$ に関しても、ブロック $B_0$ と同様に、式(1)が成り立つため、全てのブロックに対して、第1の付加情報を出力すれば良い。4つのブロックに対する付加情報の列(補間データ)は“1111”の4ビットとなる。

【0046】

従って、図6(a)の場合、縮小画像は $2 \times 2$ 画素となり、縮小画像のデータサイズは $4 \times 8$ ビットである。従って、縮小画像とその付加情報のデータ量は $4 \times 8 + 4 = 36$ ビットとなる。オリジナル画像が $128$ ビット( $= 4 \times 4 \times 8$ )であるから、 $36 / 128 = 9 / 32$ に圧縮されることになる。

【0047】

一方、図6(b)の画像データの場合、ブロック $B_0$ に着目すると、 $B_0(0, 0) \quad B_0(0, 1) \quad B_0(1, 0) \quad B_0(1, 1)$ であり、式(1)は成立しない。

【0048】

10

20

30

40

50

しかし、 $B_0(0, 1) = B_1(0, 0)$ 、 $B_0(1, 0) = B_2(0, 0)$ 、 $B_0(1, 1) = B_3(0, 0)$ を満たす。つまり、式(2)を満たす。従って、着目ブロック $B_0$ に対しては第2の付加情報(“01”の2ビット)を出力する。

【0049】

続いて、着目ブロックが $B_1$ へ移った場合について考察する。この場合、着目ブロック $B_1$ について、 $B_1(0, 0) = B_1(0, 1)$ 、 $B_1(1, 0) = B_1(1, 1)$ であるので、処理を周囲画素との一致/不一致判定へ移すことになる。すなわち、式(2)について検証することになる。

【0050】

着目ブロック $B_1$ は、画像の右端に位置するため、周囲画素を参照する場合には、予め定められた画素値(実施形態では“255”)との比較を行う。比較を行った結果、 $B_1(0, 1) = 255$ 、 $B_1(1, 0) = B_3(0, 0)$ 、 $B_1(1, 1) = 255$ であるため、第2の付加情報では再現できない。そのため、第3の付加情報を出力する。すなわち、第3の付加情報として、2ビット“00”と、それに後続して3画素分の画素データを示す24ビットのデータが出力される。

【0051】

ブロック $B_2$ 、 $B_3$ についても同様の処理を行う。ブロック $B_2$ は式(1)、(2)のいずれも満たさないため、第3の付加情報が出力される。そして、ブロック $B_3$ は式(1)を満たすので、第1の付加情報が出力される。

【0052】

図6(b)の場合、全てのブロックを処理した結果、補間データ生成部203が出力する補間データは図7に示すようなデータ構造となる。図7に示したデータ構造はあくまでも一例であり、補間データには、少なくとも、1ブロックに対して、第1、第2、第3いずれかの付加情報が含まれていればよい。

【0053】

格納部204には、上記のように、解像度変換部202により生成された縮小画像データ、及び、補間データ生成部203によって生成された補間データが格納される。

【0054】

画像展開部205は、格納部204に格納されている、縮小画像データと、補間データから、オリジナル解像度の画像の復元処理を行う。画像の展開は、補間データの生成処理と逆の処理を行えばよい。つまり、補間データを解析し、解析した付加情報から、次の3つのいずれかに基づき、縮小画像の着目画素からブロック内の未知の3画素 $X_a$ 、 $X_b$ 、 $X_c$ を得る。

(1) 縮小画像中の着目画素を単純拡大して得る。

(2) 縮小画像中の着目画素の隣接画素から得る。

(3) 補間データ中に含まれている、画素値から得る。

【0055】

具体的には、縮小画像中の着目画素 $P(x, y)$ に対する付加情報が“1”であった場合には、縮小画像の着目画素をそのまま画素 $X_a$ 、 $X_b$ 、 $X_c$ として使用してブロックを復元する。

【0056】

また、縮小画像中の着目画素 $P(x, y)$ に対する付加情報が“01”であった場合には、画素 $X_a$ を画素 $P(x+1, y)$ で再現し、画素 $X_b$ を画素 $P(x, y+1)$ で再現し、画素 $X_c$ を画素 $P(x+1, y+1)$ で再現する。ただし、着目画素 $P(x, y)$ が画像の右端に位置する場合、 $X_a = 255$ 、 $X_c = 255$ である。また、着目画素 $P(x, y)$ が画像の下端に位置する場合、 $X_b = 255$ 、 $X_c = 255$ である。特に、着目画素 $P(x, y)$ が画像の右下隅に位置する場合、 $X_a = X_b = X_c = 255$ である。

【0057】

そして、縮小画像中の着目画素 $P(x, y)$ に対する付加情報が“00”であった場合には、その付加情報に後続する3バイト(24ビット)が、画素 $X_a$ 、 $X_b$ 、 $X_c$ の順に

10

20

30

40

50

並んでいるものして再現する。

【 0 0 5 8 】

上記の処理を繰り返すことで、オリジナル解像度の画像データが復元できる。復元されたオリジナル解像度の画像は、画像出力部 2 0 6 へ入力される。

【 0 0 5 9 】

画像出力部は、画像展開部 2 0 5 から出力された画像を受け取り、本画像処理装置の外部（例えばプリンタや表示装置）へ復元画像データを出力する。

【 0 0 6 0 】

以上のように、本実施形態の画像処理装置では、入力画像を解像度変換すると同時に、縮小画像からオリジナル解像度の画像を復元するための補間データを生成する。本実施形態の符号化処理によれば、オフィス系の文書画像（文字や線画のみの画像）に対して、補間データのデータ量を少なく、簡易に生成することが可能であり、オリジナルの解像度の画像を完全に可逆で復元可能である。

10

【 0 0 6 1 】

また、実施形態では、画像の符号化と復号の両方の機能を持つ 1 つの装置に適用する例を示したが、これらは別々の装置であっても構わない。画像符号化装置を単体に適用する場合、縮小画像データと補間データとを 1 つのファイルにして出力すればよい。復号する単体の装置は、そのファイルから先ず縮小画像をメモリに展開し、その後で補間データに従ってオリジナルの画像を復元（復号）すればよい。これ以上についての説明は不要であろう。

20

【 0 0 6 2 】

以上で第 1 の実施形態の説明を終えるが、本実施形態ではオリジナル画像からサンプリングによって、水平、垂直方向とも 1 / 2 の縮小画像を生成し、その際に補間データを生成した。この縮小画像をオリジナル画像として再帰的に入力することを繰り返してもよい。この場合、オリジナル画像の第 1 段階の縮小画像と第 1 段階の補間データが得られる。そして、その第 1 段階の縮小画像から第 2 段階の縮小画像（オリジナルに対して水平、垂直とも 1 / 4 のサイズ）と第 2 段階の補間データが得られ、...、第 n 段階の縮小画像と第 n 段階の補間データが得られる。結局のところ、第 n 段階の縮小画像 + 第 n 段階の補間データ + 第 n - 1 段階の補間データ + ... + 第 1 段階の補間データが生成されることになり、いくつもサイズの画像を可逆再現可能な符号化データを生成することができることになる点で着目される。

30

【 0 0 6 3 】

なお、上記実施形態では、縮小画像の生成のためのサンプリング対象をブロックの左上隅の画素とする例を説明したが、サンプリング対象が 2 × 2 画素のいずれの位置であっても良い。

【 0 0 6 4 】

第 1 の付加情報については説明するまでもないであろう。第 2 の付加情報を生成するとき、仮に、上記と同様に、着目ブロック中の右下隅の画素 X をサンプリング対象にし、残りの 3 画素を X a , X b , X c と表わしたとする。そして、画素 X a を復元するために参照する隣接ブロック内の前記画素 X a に隣接するサンプリング対象となる画素を X 1、画素 X b を復元するために参照する隣接ブロック内の前記画素 X b に隣接するサンプリング対象となる画素を X 2、画素 X c を復元するために参照する隣接ブロック内の前記画素 X c に隣接するサンプリング対象となる画素を X 3 と定義したとき、

40

X a = X 1 且つ、

X b = X 2 且つ、

X c = X 3

を満たす場合、第 2 の付加情報を生成するようにすればよい。

【 0 0 6 5 】

[ 第 1 の実施形態の変形例 ]

上記第 1 の実施形態と同等の処理を、コンピュータプログラムによって実現する例を第

50

1の実施形態の変形例として以下に説明する。

【0066】

図14は、本変形例における情報処理装置（例えばパーソナルコンピュータ）のブロック構成図である。

【0067】

図中、1401はCPUで、RAM1402やROM1403に記憶されているプログラムやデータを用いて本装置全体の制御を行うと共に、後述する画像符号化処理、復号処理を実行する。1402はRAMで、外部記憶装置1407や記憶媒体ドライブ1408、若しくはI/F1409を介して外部装置からダウンロードされたプログラムやデータを記憶する為のエリアを備える。また、RAM1402は、CPU1401が各種の処理

10

【0068】

1406は表示装置で、CRTや液晶画面などにより構成されており、画像や文字などの情報を表示することができる。1407はハードディスクドライブ装置等の大容量の外部記憶装置である。この外部記憶装置1407には、OS（オペレーティングシステム）や後述する画像符号化、復号処理の為のプログラム、符号化対象の画像データ、復号対象画像の符号化データなどがファイルとして保存されている。また、CPU1401は、これらのプログラムやデータをRAM1402上の所定のエリアにロードし、実行すること

20

【0069】

1408は記憶媒体ドライブで、CD-ROMやDVD-ROMなどの記憶媒体に記録されたプログラムやデータを読み出してRAM1402や外部記憶装置1407に出力するものである。なお、この記憶媒体に後述する画像符号化、復号処理の為のプログラム、符号化対象の画像データ、復号対象の画像の符号化データなどを記録しておいても良い。この場合、記憶媒体ドライブ1408は、CPU1401による制御によって、これらのプログラムやデータをRAM1402上の所定のエリアにロードする。

【0070】

1409はI/Fで、このI/F1409によって外部装置を本装置に接続し、本装置と外部装置との間でデータ通信を可能にするものである。例えば符号化対象の画像データや、復号対象の画像の符号化データなどを本装置のRAM1402や外部記憶装置1407、あるいは記憶媒体ドライブ1408に入力することもできる。1410は上述の各部を繋ぐバスである。

30

【0071】

上記構成において、本装置の電源がONになると、CPU1401はROM1403のブートプログラムに従って、外部記憶装置1407からOSをRAM1402にロードする。この結果、キーボード1404、マウス1405の入力が可能となり、表示装置1406にGUIを表示することが可能になる。ユーザが、キーボード1404やマウス1405を操作し、外部記憶装置1407に格納された画像符号化処理用アプリケーションプログラムの起動の指示を行なうと、CPU1401はそのプログラムをRAM1402にロードし、実行する。これにより、本装置が画像符号化装置として機能することになる。この場合の符号化アプリケーションは、指定された画像データから先に説明したブロック単位に1画素をサンプリングして縮小画像を生成する処理と、補間データ生成処理（図1のフローチャート）に示した処理を行なう。そして、その結果を1つのファイルとしてハードディスク等の記憶装置に格納したり、外部装置もしくはネットワーク上に転送すればよいであろう。

40

【0072】

[第2の実施形態]

上記第1の実施形態では、サンプリングを行なう単位を2×2画素で構成されるブロッ

50

クとし、選択的に付加情報を決定し、補間データを生成するものであった。この場合、最小の場合に、3画素に対して1ビットの符号語(第1の付加情報)が割り当てられる。第2の実施形態では、第1の実施形態を更に発展させ、更に符号化効率を上げる例を説明する。

#### 【0073】

なお、本第2の実施形態においても、符号化対象の画像データは、説明を簡単にするために、モノクロ多値画像データとし、1画素当たり8ビット(256階調)として説明する。ただし、符号化対象の画像データは複数の色成分(RGBやYMK、Lab)でも良く、1つの成分を表わすビット数も8ビットに限らない。複数色成分が存在する場合には、各成分について同様の処理を行えば良いだけである。

10

#### 【0074】

また、本第2の実施形態における「ブロック」とは、第1の実施形態と同様に2×2画素を表わす。そして、「タイル」とは、水平方向にM個、垂直方向にN個(ともに2以上の整数)のブロックのサイズであるものとする。説明を簡単なものとするため、1タイルはM=N=8とする。すなわち、1タイルは16×16画素のサイズであるものとする。また、説明を簡単なものとするため、符号化対象の画像データの水平方向の画素数を16W、垂直方向の画素数を16H、つまり、共に16の整数倍であるものとして説明する。

#### 【0075】

図8に、本第2実施形態に係る画像符号化装置のブロック図を示す。図2と異なる点は、補間データ生成部203の前にタイル分割部801を設け、補間データ生成部203の後に、付加情報統合部802を設けた点である。これ以外は第1の実施形態と同じであるので、その説明は省略する。

20

#### 【0076】

タイル分割部801は画像入力部201から入力した画像データを16×16画素のタイルに分割し、そのタイルの画像データを補間データ生成部203に出力する。なお、補間データ生成部203は、タイル内の補間データを生成する際に、タイルの外側画素データを参照するので、タイル分割部801は、タイルの右側に1列分、タイルの下端に1ライン分だけ多い画素データを補間データ生成部203に出力する。そして、この1列、1ラインにおける画素データは、タイル分割部801がタイル単位の出力にてオーバーラップする部分でもある。

30

#### 【0077】

補間データ生成部203は、2×2画素を単位に入力し、第1実施形態で説明した手順で、補間データの生成の生成処理を行なう。

#### 【0078】

付加情報統合部802は、補間データ生成部203からの符号化データ入力部として機能するものである。そのため、付加情報統合部802は内部にバッファメモリを有する。そして、付加情報統合部802は、補間データ生成部203で生成された符号化データ(補間データ)を入力し、その内部バッファに一時的に格納し、1タイル分、すなわち、8×8個の補間データを解析し、補間データの統合処理を行なう。以下、この付加情報統合部802の処理内容をより詳しく説明する。

40

#### 【0079】

まず、付加情報統合部802は、入力された着目タイルに内包される8×8個の付加情報を解析する。そして、その解析結果に基づき、以下に示す第1統合処理、第2統合処理を実行する。

#### 【0080】

##### <第1の統合処理>

この処理は、着目タイル内の8×8個の付加情報が互いに等しい場合の処理である。付加情報の種類は、第1の実施形態で説明したように、第1、第2、第3の付加情報の3種類である。そこで、8×8個の付加情報が互いに等しい場合、それらを統合する。例えば、着目タイル内の8×8個の付加情報が同じである場合には、先頭に互いに等しいことを

50



示す1ビットの識別ビット“1”を出力する。そして、それに後続して付加情報を出し、着目タイルの統合処理を終える。従って、この識別ビットは統合できたか否かを示すフラグ情報ということもできる。

【0081】

従って、もし着目タイル内の8×8個のブロックの付加情報が第1の実施形態における第1の付加情報である“1”であった場合には、着目タイルの統合結果は2ビットの“11”となる。また、着目タイル内の8×8個のブロックの付加情報が第1の実施形態における第2の付加情報である“01”であった場合には、着目タイルの統合結果は3ビットの“101”となる。そして、着目タイル内の全ブロックが第3の付加情報である場合には、着目タイルの統合結果は3ビットの“100”となる。

10

【0082】

ここで、着目タイル内の全ブロックの付加情報が第3の付加情報であった場合、統合結果を示す3ビット“100”に後続して、各ブロック内の画素値を出力する(各ブロックの付加情報は不要である)。実施形態では、1タイル=8×8個のブロックとしている。各ブロックの復元対象の画素は3つであるので(図4参照)、8×8×3画素(バイト)が後続することになる。

【0083】

上記の如く、着目タイル内の8×8個のブロックの付加情報が全て等しい場合には、上記の処理で、着目タイルの付加情報の統合処理を終える(以降に説明する第2統合処理は行なわない)。

20

【0084】

一方、着目タイル内の8×8個の付加情報中、1つでも異なるものがあつた場合には、その旨を示す識別ビット“0”を出力し、次に説明する第2統合処理を行なう。

【0085】

<第2の統合処理>

この第2統合処理は、タイル内の全付加情報中に、少なくとも1つの付加情報が他と異なる場合の処理である。換言すれば、上記タイルのサイズより小さければ、その小さい領域にて付加情報を統合できる可能性があることになる。そこで、本第2の実施形態では、1つのタイルには8×8個の付加情報が含まれることに着目し、これを水平方向に並ぶ8つの付加情報を単位に統合処理を行なうようにした。以下、この水平方向に並ぶ8つの付加情報をライン付加情報という。

30

【0086】

本第2の統合処理では、着目ライン付加情報に含まれる8つの付加情報が互いに等しいか否かを判定する。そして、互いに等しい場合には、それを識別する1ビットの識別ビット“1”を出力し、その後、互いに等しいと判定された付加情報を1つ出力する。一方、着目ライン付加情報に含まれる8つの付加情報中、1つでも異なるものが存在した場合、1ビットの識別ビット“0”を出力し、それに後続して8つの付加情報を出力する。

【0087】

着目タイルは、8つのライン付加情報で構成されるものであるので、上記処理を最上段から最下段に向かって8回行なえばよい。

40

【0088】

以上本第2の実施形態における付加情報統合部802の処理を説明した。

【0089】

図9は、本第2の実施形態における付加情報統合部802の統合処理手順を示すフローチャートである。以下、図9のフローチャートに従って、本実施形態の付加情報統合部801の処理を説明する。

【0090】

まず、補間データ生成部203で生成された着目タイルに含まれる8×8個の付加情報を入力する(ステップS901)。

【0091】

50

次いで、入力された付加情報を解析し、 $8 \times 8$ 個の全付加情報（全符号化データ）が同じ種類（1種類）であるか否かを判定する（ステップS902）。判定の結果、全て同じ種類の付加情報である場合（YES）は、ステップS903へ処理を遷移させる。ステップS903は、着目タイルの付加情報を1つに統合したことを示す1ビットのフラグ“1”を出力する。そして、ステップS904にて、そのフラグに後続して、1つの付加情報を出力する。なお、全て同じと判断された付加情報が第3の付加情報である場合には、第3の付加情報に後続して、 $8 \times 8 \times 3$ 個の画素データを出力する。

【0092】

一方、ステップS902にて、NOと判定された場合、すなわち、着目タイルに含まれる $8 \times 8$ 個の付加情報中に、少なくとも2種類の付加情報が存在する場合、処理はステップS905に進む。このステップS905では、着目タイルの付加情報の非統合を示す1ビットのフラグ“0”を出力する。そして、ステップS906乃至S911にて、着目タイルよりも更に小さいサイズを単位とする統合処理を行なう。実際では、このサイズを水平方向に並ぶ8個の付加情報で構成されるブロックラインについてのブロックライン統合処理である。

10

【0093】

まず、ステップS906では、着目ライン付加情報（初期段階では着目タイルの最上位のラインの付加情報）に含まれる8つの付加情報が全て同じであるか否かを判断する。同じであった場合には、そのラインの付加情報を統合できるので、ステップS907にて統合されたことを示す1ビットのフラグ“1”を出力し、ステップS908にて、同じであると判定した付加情報を1つ出力する。なお、同じであると判定された付加情報が第3の付加情報である場合には、その付加情報に後続して $8 \times 3$ 画素分の画素データを出力する。

20

【0094】

また、ステップS906にて、着目ライン（初期段階では着目タイルの最上位のライン）の全付加情報中に、1つでも他と異なる付加情報が存在したと判断した場合は、ステップS909に進む。このステップS909では、そのラインの付加情報を統合できなかったことを示す1ビットのフラグ“0”を出力する。そして、ステップS910にて、8つの付加情報を出力する（第3の付加情報が存在する場合には、その第3の付加情報に後続して3個の画素データが並ぶ）。

30

【0095】

ステップS911では、着目タイルの最終ラインまで統合処理を終えたか否かを判断し、否の場合には、次のラインの統合処理を行なうため、ステップS906に処理を戻す。

【0096】

上記のようにして着目タイルについての統合処理を終えると、ステップS912にて、全タイルについての統合処理を終えたか否かを判断し、否の場合には、ステップS1に戻りつぎのタイルの付加情報についての統合処理を行なう。

【0097】

ここで1つのタイルの付加情報の統合処理の例を図10（a）乃至（c）に従って説明する。

40

【0098】

図10（a）は1つのタイルの画像データを示している。破線で囲まれた1つの格子が1画素を示している。

【0099】

図示では、第1ラインから第9ラインまでの領域が同じ値の白画素で構成されていることを示している。また、第10、第11ラインの領域が同じ値の黒画素で構成され、第12ラインから第16ラインまでの領域は同じ値の白画素で構成されている例を示している。図示の例は、文字や線画の画像の場合に多く見られるパターンでもある。

【0100】

この図10（a）のタイルの画像データを入力したときに、補間データ生成部203が

50

生成する各ブロックの付加情報の値を示すのが図10(b)である。

【0101】

図10(b)の通り、このタイル内の全付加情報は同じではない。従って、図10(c)に示すように、タイル統合が不可であったことを示すフラグ“0”を出力する(図9のステップS905)。

【0102】

次に、最初のライン付加情報を見ると、8個の付加情報は“1”である。従ってライン統合が可能であるので、フラグ“1”を出力し、その後付加情報“1”を出力する。以下、これを繰り返すことになる。

【0103】

図10(a)のタイル画像の場合、結局のところ、このタイルに対する付加情報統合部801が出力する付加情報の統合結果は、図10(c)に示す通りとなり、つぎのようなビット配列となる。

“0;11;11;11;11;101;101;11;11”

なお、ここでセミコロンは、理解を容易にするために、ライン付加情報の統合の区切りを示したものであり、実際は存在しない点に注意されたい。

【0104】

図10(a)のタイルには8×8個のブロックが含まれ、各ブロックには復元すべき3画素が存在することになるので、1タイル内に含まれる復元すべき画素数は8×8×3画素存在する。すなわち、1タイルにおける復元すべき画素の総ビット数は8×8×3×8

ビット=1536ビットである。

【0105】

図10(b)の付加情報の場合、1ビットの付加情報が全部で48個、2ビットの付加情報が全部で16であるので、1タイルの付加情報の総ビット数は48+16×2ビット=80ビットである(第1の実施形態に相当する)。

【0106】

そして、図10(c)の場合、すなわち、第2の実施形態における1タイルの付加情報の統合後の総ビット数はフラグビットを含めて、19ビットとなり、大幅に補間データのデータ量を削減可能であることがわかる。

【0107】

以上説明したように本第2の実施形態によれば、補間データ生成部で生成された、2×画素のブロック毎の付加情報に対して、予め定められた単位毎に連続しているか否かの判定を行う。付加情報が連続していれば、連続している付加情報を統合することで、補間データのデータ量を大幅に削減することが可能である。また、付加情報が連続していなければ、更に小さい単位毎に連続性を判定し、どのように統合処理を行なう。また、付加情報の連続性を調査するだけなので、統合処理のために、画像を再度スキャンする必要はなく、補間データ生成の処理時間が低下することはない。

【0108】

なお、本第2の実施形態と同等の処理をコンピュータプログラムでもって実現できることは、先に説明した第1の実施形態の変形例から明らかであるので、その説明については省略する。

【0109】

[第3の実施形態]

上記第1、第2の実施形態では、1つのブロック(2×2画素)に対し、第1と第2と第3の付加情報のいずれか1つを割り当て、そのブロックの補間データを生成した。本第3の実施形態では、付加情報の種類を3種類に限定しないで、補間データを生成する例を説明する。

【0110】

なお、本第3の実施形態でも、符号化処理対象の画像データはモノクロ多値画像データとするが、RGB画像データや、CMYKカラー画像データに適用しても良い。また、画

10

20

30

40

50

像は水平方向が16W画素、垂直方向が16H画素により構成されるものとする（水平、垂直とも16の整数倍とする）。

【0111】

図11に、本第3の実施形態に係る画像符号化装置のブロック構成図を示す。第1、第2の実施形態で説明した図1、図8と共通するブロックについては同じ番号を付し、その説明については省略する。

【0112】

以下、第2の実施形態と動作の異なる部分について説明する。

【0113】

先に説明した第2の実施形態では、2×2画素のブロックに対する付加情報を生成した後、付加情報の統合処理を行う場合の説明を行った。これに対し、本第3の実施形態においては、付加情報の生成の単位を適応的に変えながら補間データを生成する方法について説明を行う。そのため、図8の付加情報統合部802はなくなり、代わりに、第1、第2の実施形態における補間データ生成部203に変わって、本第3の実施形態の補間データ生成部1101を設けた。

10

【0114】

図12は、本第3の実施形態に係る、補間データ生成部1101のブロック構成図を示す。この補間データ生成部1101はフラット判定部1201、ノンフラットブロック解析部1202で構成される。

【0115】

なお、以下の説明で縮小画像中の画素Xから、Xa, Xb, Xc（図4参照）として復元できるブロックを“フラットブロック”と呼ぶ。換言すれば、ブロック内の2×2画素が互いに等しい値（同色）であるブロックをフラットブロックと呼ぶ。また、ブロック内の2×2画素の中に、他とは異なる画素値を持つものが少なくとも1つ存在するブロックをノンフラットブロックと呼ぶ。そして、タイル内の全ブロックがフラットブロックである場合、そのタイルをフラットタイルと呼ぶ。そして、タイル内に少なくとも1つのノンフラットブロックが存在する場合、そのタイルをノンフラットタイルと呼ぶことにする。また、1つのタイル中の水平方向に並ぶ8つのブロックをブロックラインと呼ぶ。そして、ブロックライン中の8つのブロックが全てフラットブロックである場合、そのブロックラインをフラットブロックラインと呼び、1つでもノンフラットブロックを含むブロックラインをノンフラットブロックラインと呼ぶこととする。また、本第3の実施形態では、タイル外の周囲画素を参照する場合には、そのタイル外の画素の値は“255”であるものとして説明する。

20

30

【0116】

フラット判定部1201は、水平方向に2M画素、垂直方向に2N画素で構成されるタイル単位に画像データを入力して符号化するタイル符号化部として機能する。説明を簡単なものとするため、M=N=8とする。そして、フラット判定部1201は、入力したタイル（16×16画素）中のそれぞれのブロック（2×2画素）に着目し、その中の縮小画像に利用された画素Xを除く他の画素Xa, Xb, Xcが、縮小画像中の画素Xから単純に復元できるか否かを判定する。そして、フラット判定部1201は、この判定結果に基づき処理を行なう。

40

【0117】

ノンフラットブロック解析部1202は、フラット判定部1201の判定結果に基づき、ブロック内に存在する色数や配置に従って付加情報を生成し、出力する。

【0118】

以下、フラット判定部1201の処理内容を説明した後、ノンフラットブロック解析部1202の処理内容を説明する。

【0119】

<フラット判定部の説明>

図13はフラット判定部1201の処理手順を示すフローチャートである。

50

## 【 0 1 2 0 】

フラット判定部 1 2 0 1 は、ステップ S 1 3 0 1 にて、1 タイル分の画像データ（タイルデータ）を入力する。そして、入力したタイルデータ中の各ブロックについてフラットブロックであるか、ノンフラットブロックであるかの判定処理を行なう。

## 【 0 1 2 1 】

ステップ S 1 3 0 2 では、入力した着目タイルがフラットタイルであるか否かを判定する（第 1 の判定処理）。入力したタイル内の 8 × 8 個の全ブロック（各ブロックは 2 × 2 画素）がフラットブロックであるか否かを判定する、と言い換えても良い。

## 【 0 1 2 2 】

着目タイルがフラットタイルであると判定した場合、縮小画像中の 8 × 8 画素の画像データから 1 6 × 1 6 画素の画像データを単純に生成することができることを意味する。従って、ステップ S 1 3 0 3 に処理を進め、着目タイルがフラットタイルであることを示す 1 ビットのフラグ “ 1 ” をノンフラットブロック解析部 1 2 0 2 に出力する。

10

## 【 0 1 2 3 】

一方、ステップ S 1 3 0 2 にて、着目タイルがノンフラットタイルであると判定した場合、処理はステップ S 1 3 0 4 に進み、先ず、ノンフラットタイルであることを示すフラグ “ 0 ” をノンフラットブロック解析部 1 2 0 2 に出力する。この後、ステップ S 1 3 0 5 以降では、着目タイル内のブロックライン単位とするブロックライン符号化処理を行なう。以下に説明するブロックライン単位の符号化処理は、着目タイルの最上位から最下位に位置するブロックラインに向かって順次処理するものである点に注意されたい。

20

## 【 0 1 2 4 】

先ず、ステップ S 1 3 0 5 では、タイルの最上位の（第 1 番目の）ブロックラインに着目し、それがフラットブロックラインであるか否かを判定する（第 2 の判定処理）。水平方向に並ぶ 8 つのブロックの全てがフラットブロックであるか否かを判定する、と言い換えても良い。

## 【 0 1 2 5 】

着目ブロックラインがフラットブロックラインであると判断した場合、縮小画像中の水平方向に並ぶ 8 つの画素データから、水平方向 1 6 画素、垂直方向 2 画素の矩形領域の画像データが単純に復元できることを意味する。従って、ステップ S 1 3 0 6 にて、着目ブロックラインがフラットブロックラインであると示すフラグ “ 1 ” をノンフラットブロック解析部 1 2 0 2 に出力する。

30

## 【 0 1 2 6 】

また、ステップ S 1 3 0 5 にて、着目ブロックラインがノンフラットブロックラインであると判定した場合、処理はステップ S 1 3 0 7 に進む。このステップ S 1 3 0 7 では、着目ブロックラインがノンフラットブロックラインであることを示すフラグ “ 0 ” をノンフラットブロック解析部 1 2 0 2 に出力する。そして、ステップ S 1 3 0 8 に処理を進める。

## 【 0 1 2 7 】

ステップ S 1 3 0 8 に処理が進んだ場合、着目ブロックラインには少なくとも 1 つのノンフラットブロックが混在していることを意味する。そこで、このステップ S 1 3 0 8 では、着目ブロックラインにおけるフラットブロック、ノンフラットブロックの並びが、着目ブロックライン以前にノンフラットブロックラインであると判定されたブロックラインのそれと一致するか否かを判定する（第 3 の判定処理）。なお、着目ブロックラインが着目タイルの最初のブロックラインである場合には、それ以前のブロックラインは存在しない。そこで、着目タイルのブロックラインの判定に先立ち、ブロックラインの全ブロックがノンフラットブロックであることを示す情報を予め用意しておく。つまり、フラットブロックを “ 1 ”、ノンフラットブロックを “ 0 ” と定義したとき、初期値として 8 個（8 ビット）の “ 0 ” を用意しておく。以降、この 8 ビットの “ 参照ノンフラットブロックライン情報 ” と呼ぶ。勿論、復号装置でも 1 タイルの復元する際には、“ 参照ノンフラットブロックライン情報 ” を用意しておく。

40

50

## 【 0 1 2 8 】

着目ブロックラインが、着目タイル内で最初にノンフラットブロックラインであると判定されたブロックラインであったとしたとする。この場合、ステップ S 1 3 0 8 では、着目ブロック内の 8 個のブロックのフラットブロック / ノンフラットブロックの判定結果の並びが、参照ノンフラットブロックライン情報と一致するか否かを判定することになる。

## 【 0 1 2 9 】

一致したと判定した場合には、ステップ S 1 3 0 8 にて、着目ブロックライン内の各ブロックの判定結果が、参照ノンフラットブロックライン情報と一致したことを示す 1 ビットのフラグ “ 1 ” をノンフラットブロック解析部 1 2 0 2 に出力する。

## 【 0 1 3 0 】

上記を図 1 0 ( b ) を参照して説明する。最初に着目ブロックラインがノンフラットブロックラインであると判定されるのは、図 1 0 ( b ) の場合、5 番目のブロックラインである。図示では、5 番目のブロックラインに含まれる 8 つの付加情報は “ 0 1 ” であり、これはノンフラットブロックを示すからである。また、この 5 番目のブロックラインに含まれる 8 個のブロックは、全てがノンフラットブロックであることを示すので、参照ノンフラットブロックライン情報 ( 初期の 8 ビット “ 0 0 0 0 0 0 0 ” ) と一致する。それ故、ステップ S 1 3 0 9 にて、1 ビットのフラグ “ 1 ” をノンフラットブロック解析部 1 2 0 2 に出力する。

## 【 0 1 3 1 】

一方、ステップ S 1 3 0 8 にて、不一致であると判定した場合、処理はステップ S 1 3 1 0 に進む。

## 【 0 1 3 2 】

このステップ S 1 3 1 0 に遷移するのは、着目ブロックラインの判定結果が、参照ノンフラットブロックライン情報と一致しない場合である。そこで、まず、不一致を示す 1 ビットのフラグ “ 0 ” を出力し、その後で、着目ブロックラインの 8 個 ( 8 ビット ) のフラットブロック / ノンフラットブロックの判定結果の情報をノンフラットブロック解析部 1 2 0 2 に出力する。そして、ステップ S 1 3 1 1 にて、着目ブロックラインのフラットブロック / ノンフラットブロックの判定結果 ( 8 ビット ) で、参照ノンフラットブロックライン情報を更新する。

## 【 0 1 3 3 】

ステップ S 1 3 0 6、S 1 3 0 9、S 1 3 1 1 のいずれかの処理を終えると、処理はステップ S 1 3 1 2 に進む。このステップ S 1 3 1 2 では、着目ブロックラインが着目タイルの最後のブロックラインであるか否かを判定する。否の場合には、ステップ S 1 3 1 3 に進み、判定対象を次のブロックラインにし、ステップ S 1 3 0 5 以降の処理を繰り返す。また、ステップ S 1 3 1 2 にて、着目ブロックラインが着目タイルの最後のブロックラインであると判定した場合には、フラット判定部 1 2 0 1 の 1 タイル分のフラット判定処理を終える。

## 【 0 1 3 4 】

< ノンフラットブロック解析部の説明 >

ノンフラットブロック解析部 1 2 0 2 は、フラット判定部 1 2 0 1 と同様に、1 タイルの相当する処理を開始時に、1 ブロックラインの 8 つのブロックの全てがノンフラットブロックを示す「参照ノンフラットブロックライン情報」を設定しておく。そして、ノンフラットブロック解析部 1 2 0 2 は、フラット判定部 1 2 0 1 からの判定結果をそのまま通過させ、格納部 2 0 4 に出力する。このとき、ノンフラットブロック解析部 1 2 0 2 は、先の図 1 3 のステップ S 1 3 1 0 にて、フラグ “ 0 ” が出力されたブロックラインが存在すると判定した場合、それに後続する各ブロックのフラットブロック / ノンフラットブロックを示す判定結果で、参照ノンフラットブロックライン情報を更新する。もし、着目ブロックラインに対し図 1 3 のステップ S 1 3 0 9 の処理が行われた場合には、参照ノンフラットブロックラインを参照することで、どのブロックがノンフラットブロックであるのかを判定できる。また、着目ブロックラインに対し、図 1 3 のステップ S 1 3 1 0 の処理

10

20

30

40

50

が行われていると判断した場合には先頭のフラグ“0”に後続する8ビットの値を調べれば、ブロックがノンフラットブロックであるのかを判定できる。すなわち、ノンフラットブロック解析部1202は、着目タイル中の全てのノンフラットブロックの位置と個数を判定できることになる。

【0135】

そして、ノンフラットブロック解析部1202は、ノンフラットブロックと判定されたブロックにつき、図15のフローチャートに従ったノンフラットブロック符号化処理を行なう。以下、同図に従って、ノンフラットブロック解析部1202の処理を説明する。

【0136】

まず、ステップS1501にて、1個のノンフラットブロックを示す2×2画素データを入力する。そして、ステップS1502にて、着目ブロックが第1の実施形態で示した式(2)を満足するか否かを判定する(第4の判定処理)。

【0137】

式(2)を満足すると判定した場合、縮小画像中の画素P(x, y)を着目ブロックの左上隅の画素Xとしたとき(図4参照)、復号装置(実施形態では画像展開部205)は、着目ブロックの残りの3つの画素Xa, Xb, Xcは、次式から復元できる。従って、ステップS1503にて1ビットのフラグ“1”を格納部204に出力する。

$$X_a = P(x + 1, y)$$

$$X_b = P(x, y + 1)$$

$$X_c = P(x + 1, y + 1)$$

【0138】

また、着目ブロックが式(2)を満足しないと判定した場合、ステップS1504にて満足しないことを示す1ビットのフラグ“0”を格納部204に出力する。

【0139】

次に、処理はステップS1505に進み、着目ブロックに含まれる色数が“2”か、それ以上(3、もしくは4色)かを判定する(第5の判定処理)。なお、着目ブロックはノンフラットブロックであるので、色数が“1”となることはない。

【0140】

着目ブロックに含まれる色数が“2”より大きい(3、もしくは4色)と判定した場合、ステップS1512に進み、その着目ブロックについては1ビットのフラグ“0”を格納部204に出力する。そして、ステップS1513にて、着目ブロック中の非使用の3画素のデータ(8×3=24ビット)を格納部204に出力する(第1の出力処理)。

【0141】

一方、着目ブロックに含まれる色数が“2”であると判定した場合、ステップS1506にて出現色数が“2”であることを示す1ビットのフラグ“1”を格納部204に出力する。そして、ステップS1507にて、着目ブロック内の、縮小像に利用されている図4の画素Xを除く3画素Xa, Xb, Xcが、図16のパターン1601乃至1607のいずれであるのかを示す画素配置情報を格納部204に出力する(第2の出力処理)。図16に示すようにパターン数は全部で7つあるので、画素配置情報は3ビットであれば1つのパターンを特定できる。ただし、実施形態では、縮小画像に利用されている画素と同じ色の画素は“1”、異なる色の画素を“0”と表わし、図4の画素Xa, Xb, Xcに相当する各ビットをその順番に出力するようにした(この場合も3ビットとなる)。例えば、図16のパターン1601に一致する場合、画素Xa, Xb, Xcは同色であり、且つ、縮小画像に利用されている画素の色とは異なるので、画素配置情報の3ビットは“000”となる。また、パターン1602の場合は、“100”、パターン1603の場合には“010”、パターン1604の場合には“001”となる。他は説明するまでもないであろう。

【0142】

次いで、ステップS1508にて、着目ブロック内の縮小画像に利用された画素Xを除く画素Xa, Xb, Xcのうち、画素Xと異なる色の画素と同じ色の画素が、近傍に存在

10

20

30

40

50

するか否かを判断する（第6の判定処理）。本実施形態では、比較する近傍の画素は、第1の実施形態で示した第2の付加情報を参照する3つの画素とした。すなわち、着目ブロックを $B_n$ 、その右隣のブロックを $B_{n+1}$ 、着目ブロックの直下のブロックを $B_{n+W}$ 、着目ブロックの右下に隣接するブロックを $B_{n+W+1}$ と表わしたとする。この場合に、仮に、着目ブロックが図16のパターン1601であるとするなら、画素 $X_a(X_b, X_c)$ でも同じ)と、 $B_n(0, 0)$ 、 $B_{n+1}(0, 0)$ 、 $B_{n+W}(0, 0)$ とを順番に比較し、一致する画素が存在するか否かを判定する。ここで、注意したい点は、画素 $B_n(0, 0)$ 、 $B_{n+1}(0, 0)$ 、 $B_{n+W}(0, 0)$ はいずれも縮小画像にそのまま利用されていることである。また、着目ブロックがパターン1603であった場合、着目ブロック中の縮小画像に利用された画素と異なる色の画素は、図4から画素 $X_b$ であるので、その画素 $X_b$ と、 $B_n(0, 0)$ 、 $B_{n+1}(0, 0)$ 、 $B_{n+W}(0, 0)$ とを順番に比較する。

10

## 【0143】

さて、一致する近傍画素が存在すると判断した場合、処理はステップS1509に進み、その一致する画素の相対位置を示す情報を格納部204に出力する（第3の出力処理）。実施形態の場合、比較対象となる近傍の画素数は“3”であるので、一致する画素の相対位置を特定する情報は2ビットあれば十分である。実施形態では、一致するのが近傍画素 $B_n(0, 0)$ であった場合には2ビットの“11”、 $B_{n+1}(0, 0)$ であった場合には2ビットの“01”、 $B_{n+W}(0, 0)$ であった場合には2ビットの“10”を格納部204に出力する。

## 【0144】

20

一方、ステップS1508にて、着目ブロック内の縮小画像に利用された画素 $X$ を除く画素 $X_a, X_b, X_c$ のうち、画素 $X$ と異なる色の画素と同じ色の画素が、近傍に存在しないと判断した場合、処理はステップS1510に進む。このステップS1510では、2ビットの“00”を格納部204に出力する。なお、2ビットの“00”は、ステップS1509では利用されていない点に注意されたい。

## 【0145】

次いで、ステップS1511にて、着目ブロック内の縮小画像に利用された画素 $X$ を除く画素 $X_a, X_b, X_c$ のうち、画素 $X$ と異なる色の画素の値（8ビット）を出力する（第4の出力処理）。図16のパターンのいずれかが特定でき、且つ、着目ブロック内の縮小画像に利用された画素 $X$ を除く画素 $X_a, X_b, X_c$ のうち、画素 $X$ と異なる色の画素の値がわかれば、復号装置は図16のパターン1601乃至1607の各画素の値を復元できることは理解できよう。

30

## 【0146】

この後、処理はステップS1514に進み、着目ブロックが着目タイル中の最後のノンフラットブロックであるか否かを判断する。否の場合にはステップS1515に進んで、次のノンフラットブロックを示す $2 \times 2$ 画素の画像データを入力し、ステップS1502に処理を戻す。

## 【0147】

また、ステップS1514にて、着目ブロックが着目タイル中の最後のノンフラットブロックであると判断した場合には、着目タイル内のノンフラットブロックの処理を終了する。

40

## 【0148】

以上説明したように本第3の実施形態によれば、補間データ生成部1101は最初はタイルがフラットタイルであるか否かを判断し、もしそうであればタイルにつき1ビットの付加情報を生成し、終了する。そして、フラットタイルではないと判断した場合には、ノンフラットタイルであることを示す1ビットを出力し、今度はタイルよりも狭いブロックラインについて検証を行なう。以下、この処理を最終的なサイズ（ブロック）に到達するまで繰り返す。この結果、縮小画像からオリジナルの解像度の画像を生成するための付加情報を、第1、第2の実施形態より更に削減することが期待できる。

## 【0149】

50



なお、実施形態では、タイル、ブロックライン、ブロックの3段階の例を説明したが、更に多段階にしても良い。

【0150】

また、本第3の実施形態に相当する処理をコンピュータプログラムでもって実現できることは、先に説明した第1の実施形態の変形例と同様、明らかである。

【0151】

[第4の実施形態]

上記第1乃至第3の実施形態では、縮小画像+補間データ(符号化データ)でもってオリジナル画像を完全に復元する技術を開示した。すなわち、第1乃至第3の実施形態は可逆符号化を開示するものであった。本第4の実施形態では、非可逆の補間データを生成する方法について説明を行う。

10

【0152】

なお、本第4の実施形態でも、処理対象の画像データは、説明を簡単にするためにモノクロ多値画像データとするが、RGB画像データや、CMYKカラー画像データに適用しても良い。

【0153】

本第4の実施形態では、ノンフラットブロック解析部1202の処理内容が第3の実施形態と異なり、それ以外については第3の実施形態と同様であるものとし、その説明は省略する。

【0154】

20

図17は、本第4の実施形態におけるノンフラットブロック解析部1202の処理手順を示すフローチャートである。図17のフローチャートは、図15のフローチャートに代わるものであると理解されたい。それ故、同処理内容について同参照符号を付した。

【0155】

先ず、ステップS1501では、ノンフラットブロックを示す $2 \times 2$ 画素の画像データを入力する。そして、ステップS1701において、入力された着目ブロック $B_n$ に対して、図4の $X_a$ 、 $X_b$ 、 $X_c$ に位置する3画素が、参照する周囲画素に近い色であるかどうかの判定を行う(第7の判定処理)。すなわち、第1乃至第3の実施形態の如く完全一致を条件とはしない。

【0156】

30

具体的には、この許容色差を示す閾値 $Th1$ (ユーザが適当に操作部やキーボード等で適宜変更できるとする)としたとき、次式を満たすとき、画素 $X_a$ 、 $X_b$ 、 $C_c$ に位置する3画素が、参照する周囲画素に近い色であると判定する。

$|B_n(0, 1) - B_{n+1}(0, 0)| \leq Th1$ 、且つ、

$|B_n(1, 0) - B_{n+W}(0, 0)| \leq Th1$ 、且つ、

$|B_n(1, 1) - B_{n+W+1}(0, 0)| \leq Th1$

(ここで、 $|z|$ は $z$ の絶対値を示している)。

【0157】

なお、発明者等は、この閾値 $Th1$ は一桁の数値であり、望ましくは $Th1 = 4$ であることを見出している。

40

【0158】

上記式を満足する場合、処理をステップS1503へ移し、フラグ“1”を出力して、着目ブロック $B_n$ の処理は終了し、次のノンフラットブロックの処理へ移る。このとき、設定するしきい値によっては、補間データ量を抑制しつつも、画質劣化が視認できない程度の画質を保持することが可能である。

【0159】

また、着目ブロック $B_n$ に出現する色数が3色または4色の場合、第3の実施形態では、3画素分の画素値を出力する必要があった。本第4の実施形態では、画素値の出力(ステップS1513)の前に、判定処理を示すステップS1702を追加した。このステップS1702は、図4の画素 $X_a$ 、 $X_b$ 、 $X_c$ に位置する3画素が、それぞれ、隣り合う

50

画素の平均値と近い値（閾値  $Th_2$  以下）であるかどうかの判定を行う。

【0160】

具体的には、先ず、次式によって縮小画像に利用される4画素の平均値  $X_{ave1}$  乃至  $X_{ave3}$  を算出する（小数点以下を切り捨てとする）。

$$X_{ave1} = \{ B_n(0, 0) + B_{n+1}(0, 0) \} / 2$$

$$X_{ave2} = \{ B_n(0, 0) + B_{n+W}(0, 0) \} / 2$$

$$X_{ave3} = \{ B_n(0, 0) + B_{n+W+1}(0, 0) \} / 2$$

【0161】

そして、次式が満足するとき、画素  $X_a$ ,  $X_b$ ,  $X_c$  に位置する3画素が、それぞれ、隣り合う画素の平均値と近い値（閾値以下）であると判定する。

$$| B_n(0, 1) - X_{ave1} | < Th_2、且つ、$$

$$| B_n(1, 0) - X_{ave2} | < Th_2、且つ、$$

$$| B_n(1, 1) - X_{ave3} | < Th_2$$

ここで、閾値  $Th_2$  もユーザが適宜設定するものとするが、大きくても一桁の正の整数が望ましい（ここでも、閾値  $Th_2 = 4$  とする）。

【0162】

判定の結果、 $X_a$ ,  $X_b$ ,  $X_c$  の位置の画素それぞれが全て平均値との差分が小さければ（YES）、ステップ S1703 へ処理を移し、フラグ "1" を出力する（第5の出力処理）。一方、判定の結果、 $X_a$ ,  $X_b$ ,  $X_c$  の位置の画素それぞれが全て平均値との差分が大きければ（NO）、ステップ S1704 へ処理を移し、フラグ "0" を出力し、ステップ S1513 で、第3の実施形態と同様に、画素値を出力する。

【0163】

以上の処理により、元の画像データとの画素値の誤差を制御しつつ、可逆の場合とほとんど処理の変更なく、非可逆の補間データを生成可能である。画素の値に或る程度の許容差を設けたことにより、第3の実施形態と比較して、補間データのデータ量が更に削減できることになることは容易に理解できよう。

【0164】

なお、本第4の実施形態では、非可逆の補間データ生成をステップ S1701 の判定と、ステップ S1702 の判定の2つを用いる例を説明したが、どちらか一方を適用することも可能である。

【0165】

[第5の実施形態]

本第5の実施形態では、可逆と非可逆の補間データ生成を、切り替えて符号量制御を行う方法について説明する。

【0166】

補間データの符号量を、ある決められたデータ量以内に納めなければならない場合、符号量と画質の関係を考慮しつつ、段階的に符号量を制御可能である。

【0167】

まず、画質最優先で、符号量を制御しなくてもよい場合は、可逆の補間データを生成する。この場合には、復元対象である3画素の画素値をそのまま出力するブロックも存在するため、必ずしもデータ量が小さくなる保障がない。

【0168】

次に、画質を保持しつつも、符号量を抑えたい場合には、第4の実施形態で説明したような、元の画像と復元した画像の画素値の最大誤差を一定値以下に抑えつつ、符号量も抑制できるような非可逆の補間データ生成方法を適用すればよい。

【0169】

更に、補間データの符号量を抑制しなければならない場合には、補間データとして画素値をそのまま出力している画素の、階調または解像度の削減を行う。例えば、 $2 \times 2$  画素のブロックが3, 4色で構成されている場合には、2色に減色または、階調を8ビットから4ビット削減する。こうすることにより、画素値を出力した場合に3画素  $\times$  8ビット分

10

20

30

40

50

の符号量であったものが、2色に減色した場合には1画素×8ビットとなり、階調を8ビットから4ビットに削減した場合には、3画素×4ビットとなる。この方法によれば、他にフラグが付加されるものの、多くても、元の画像の半分近くまで、補間データの符号量を削減することが可能となる。

【0170】

また、別の制御方法として、2×2画素のブロックが3, 4色で構成されている場合には、画素値の出力をしていたところを、画素値の出力をしないことで、補間データの符号量制御が可能である。この場合には、画像展開時に、縮小画像の1画素を単純拡大する。または、3, 4色で構成されるブロックであると判定されたときに、2×2画素の平均値を算出し、縮小画像の画素を平均値に置き換えておき、画像展開時には、平均値に置き換えられた縮小画像の1画素を単純拡大することで、画像を復元することが可能である。この場合、2×2画素が1色になるため、画像の一部分のみ解像度が落ちた状態になる。また、2×2画素のブロック内で3, 4色存在するような領域は、高解像度自然画のような領域である可能性が高いと考えられ、解像度を落としても、それほど画質劣化は目立たないような箇所であると考えられる。

【0171】

以上の説明では、補間データを生成する際に、画質優先であるか、符号量優先であるかによって、別の方法として説明したが、可逆で補間データを生成しておいて、一部のデータを切り捨てることによって、符号量を制御することが可能となる。例えば、3, 4色で構成されるブロックに対して、可逆の補間データを生成する場合には、画素 $X_a$ ,  $X_b$ ,  $X_c$ の位置の画素値を出力するが、補間データ生成の途中や、補間データ生成が終了後に3画素分の画素値を切り捨てることで制御できる。この場合、画素値を切り捨てたブロックは、画像展開時に縮小画像の画素を用いて単純拡大すればよい。また、可逆の補間データ作成と同時に、情報を切り捨てた際に必要になるデータを生成しておけば、補間データ生成をやり直すことなく、符号量の少ない別の情報に切り替えることも可能である。例えば、3, 4色で構成されるブロックに対して、可逆の補間データを生成する場合には、画素 $X_a$ ,  $X_b$ ,  $X_c$ の位置の画素値を出力するが、同時に、2×2画素の平均値を保持しておく。そして、補間データ生成の途中や、補間データ生成が終了したあとで、符号量制御が必要になった場合には、画素 $X_a$ ,  $X_b$ ,  $X_c$ の位置の画素値を補間データから切り捨て、算出しておいた平均値に置き換える。以上のような方法により、補間データ生成の途中や補間データ生成後に補間データの一部を切り捨てたり、データ量の少ない情報に置き換えたりすることで、簡単に符号量の制御が可能である。

【0172】

[第6の実施形態]

第1乃至第5の実施形態では、オリジナル画像から縮小画像を生成する際、2×2画素の左上に位置する画素 $B_n(0, 0)$ を縮小画像の画素として用い、他の3画素 $B_n(1, 0)$ 、 $B_n(0, 1)$ 、 $B_n(1, 1)$ の補間データの生成を行った。しかし、縮小画像の利用する画素は必ずしも2×2画素のブロックの左上隅の画素でなくても構わず、 $B_n(1, 0)$ 、 $B_n(0, 1)$ 、 $B_n(1, 1)$ のいずれであっても構わない。本第6の実施形態では、2×2画素のブロック $B_n$ 内の、右下に位置する画素 $B_n(1, 1)$ を縮小画像の画素とする場合について説明を行う。

【0173】

本第6の実施形態では、図18に示す、2×2画素のブロック $B_n$ に着目すると、 $X'$ の位置の画素を縮小画像の画素とし、他の $X_{a'}$ ,  $X_{b'}$ ,  $X_{c'}$ 3画素が復元対象となる。

【0174】

基本的には、左上画素をサブサンプリングするのと、右下画素をサブサンプリングしたのとでは、点对称に逆の位置を参照すればよい。

【0175】

例えば、先に説明した各実施形態では、着目ブロック内の画素 $B_n(0, 1)$ に対して

10

20

30

40

50

は  $B_{n+1}(0, 0)$ 、 $B_n(1, 0)$  に対して  $B_{m+w}(0, 0)$ 、 $B_n(1, 1)$  に対しては  $B_{n+w+1}(0, 0)$  であった。

【0176】

これに対し、本第6の実施形態では、図18に示す丸印の位置の画素、つまり、 $B_n(1, 0)$  に対しては  $B_{n-1}(1, 1)$  を参照し、 $B_n(0, 1)$  に対しては  $B_{n-w}(1, 1)$ 、 $B_n(0, 0)$  に対しては  $B_{n-w-1}(1, 1)$  を参照すればよい。こうすることで、第1から第5の実施形態と同様の方法で、補間データの生成は可能である。

【0177】

従って、第5の実施形態におけるステップS1701を別な表現で示すのであれば次の通りである。

【0178】

まず、着目ブロック内のサンプリング対象の画素を  $X$ 、着目ブロック内の非サンプリング対象の3つの画素を  $X_a$ 、 $X_b$ 、 $X_c$  と定義する。そして、着目ブロックに隣接するブロック中の、画素  $X_a$  に隣接するサンプリング対象となる画素を  $X_1$  を定義する。同様に、着目ブロックに隣接するブロック中の、画素  $X_b$  に隣接するサンプリング対象となる画素を  $X_2$ 、着目ブロックに隣接するブロック中の、画素  $X_c$  に隣接するサンプリング対象となる画素を  $X_3$  と定義しと定義する。そして、予め設定された正の閾値を  $Th_1$  とする。この場合、ステップS1701は、

$$\begin{aligned} |X_a - X_1| & \leq Th_1 \quad \text{且つ、} \\ |X_b - X_2| & \leq Th_1 \quad \text{且つ、} \\ |X_c - X_3| & \leq Th_1 \end{aligned}$$

を満たすか否かを判定すれば良い。この条件を満たすとき、着目ブロック内の非サンプリングの3画素が、着目ブロックに隣接する3つのブロックのサンプリング対象となる画素から復元できると判定する。

【0179】

同様に、図17のステップS1702は、上記と同じ定義とするなら、閾値  $Th_2$  に対して次の条件を満たすか否かを判定する。

$$\begin{aligned} |X_a - (X + X_1) / 2| & \leq Th_2、\text{且つ、} \\ |X_b - (X + X_2) / 2| & \leq Th_2、\text{且つ、} \\ |X_c - (X + X_3) / 2| & \leq Th_2 \end{aligned}$$

【0180】

第1から第5の実施形態のように、左上画素  $X$  をサブサンプリングして、着目ブロックの右側と下側の画素を参照する場合には、画像展開時に、まだ復号処理していない位置の画素は参照できないため、縮小画像の画素しか参照できない。そのため、縮小画像の画素に当たる、 $B_{n+1}(0, 0)$ 、 $B_{n+w}(0, 0)$ 、 $B_{n+w+1}(0, 0)$  の位置の画素を先読み必要があった。しかし、本第6の実施形態によれば、右下画素  $X'$  をサブサンプリングして、着目ブロック  $B_n$  の上側と左側とを参照する場合には、すでに画像が展開してあるブロックであるので、縮小画像の画素に当たる  $B_{n-1}(1, 1)$ 、 $B_{n-w}(1, 1)$ 、 $B_{n-w-1}(1, 1)$  に位置する画素（図18の丸印の位置）以外も参照可能である。例えば、第2の付加情報を生成する場合や、2色で構成されているブロックに対して、2色目の色と同じ色が周辺画素に存在するか否かの判定を行う場合には、参照画素を図18の丸印の位置画素に限定する必要はない。もちろん、丸印の位置のみを参照にしてもいいし、星印の位置を参照画素にしても構わない。また、丸印と星印の両方を参照画素にしてもよい。

【0181】

一致 / 不一致判定で参照できる範囲を変更または、拡張できれば、補間データ生成時の符号化効率が向上する可能性がある。

【0182】

[第7の実施形態]

本第7実施形態では、ロスレスの補間データ生成時に取得した統計情報を利用して、ロッシェの補間データ生成のパラメータを決定することで、効率よく補間データ量を目標符

10

20

30

40

50

号量に抑えるような符号量制御の方法について説明を行う。

【 0 1 8 3 】

図 1 9 は、本第 7 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。

【 0 1 8 4 】

ステップ S 1 9 0 1 のロスレス補間データ生成の方法はこれまで説明したとおりであるので、本実施形態では説明を省略するが、ロスレス補間データ生成時に取得する統計情報について、以下に説明を行う。

【 0 1 8 5 】

ロスレス補間データ生成では、縮小画素や注目ブロックの周囲画素から、補間対象画素が再現できない場合には、補間対象画素値を符号化データとして出力する（図 1 5 に示したノンフラットブロック処理のステップ S 1 5 1 3、S 1 5 1 1 参照）。画素値を出力した場合には、先にも説明したように、画素値以外にフラグ情報を持つために、オリジナルよりもデータ量が大きくなってしまふ場合がある。そのため、ロッキーの補間データ生成の際は、画素値出力するブロックを減少させるか、削除することがデータ量の削減効果が高いと言える。

10

【 0 1 8 6 】

そこで、ロスレスの補間データ生成時に、どれくらいのブロックが画素値出力の処理に入るか、カウントしておくことにする。この画素値出力したブロックのカウント数を統計情報として保持しておく。

【 0 1 8 7 】

ロスレスの補間データ生成と統計情報の記録が終了したら、ステップ S 1 9 0 2 にロッシ - 符号化の為のパラメータ P r m をゼロクリアし、ステップ S 1 9 0 3 に処理を進める。

20

【 0 1 8 8 】

ステップ S 1 9 0 3（第 1 の比較工程）は、ロスレス補間データ生成で生成された補間情報のデータ量と、予め設定された目標データ量の比較を行う。この比較結果が、補間データ量が目標値よりも大きいことを示す場合、ステップ S 1 9 0 4 へ処理を進める。また、目標値以下であることを示す場合には、ステップ S 1 9 0 8 へ処理を進めて、ロスレスの補間データを出力する。

【 0 1 8 9 】

ステップ S 1 9 0 4（第 2 の比較工程）では、ロスレス補間生成で取得した統計情報である、画素値出力回数  $i$ （画素値を出力したブロック数）と、予め設定しておいた閾値  $T h 3$  の比較を行う。閾値  $T h 3$  は、任意の値を設定すればよいが、本実施形態では、入力画像全体の  $2 \times 2$  画素のブロック数の 30% を閾値  $T h 3$  として設定する。例えば、A 4、600 dpi の画像であれば、画像サイズは約  $4,960 \times 7,016$  pixel であり、 $2 \times 2$  画素の総ブロック数は 8,699,840 個となる。この 30% を閾値  $T h 3$  とすると、 $T h 3$  は 2,609,952 となる。そして、この回数  $i$  と閾値  $T h 3$  とを比較し、その比較結果に従って、後述する減色の割合を設定するパラメータ決定する。具体的には、画素出力回数  $i$  が閾値  $T h 3$  よりも大きい場合（YES）は、処理をステップ S 1 9 0 5 へ進め、ロッキー補間生成で使用するパラメータ P r m に P r m 1 を加算することで更新する。一方、画素値出力回数  $i$  が閾値  $T h 3$  以下の場合（NO）は、処理をステップ S 1 9 0 6 へ進め、ロッキー補間生成で使用するパラメータ P r m に P r m 2 を加算し、更新する。但し、 $P r m 1 > P r m 2$  とする。

30

【 0 1 9 0 】

ステップ S 1 9 0 5 又は、ステップ S 1 9 0 6 でパラメータ P r m の更新を行った後、ステップ S 1 9 0 7 へ処理を進め、ロッキー補間データの生成を行う。本実施形態におけるロッキー補間生成方法の詳細について以下に説明を行う。

40

【 0 1 9 1 】

図 2 0 に、ステップ S 1 9 0 7 のロッキー補間生成処理を行なうブロック構成図を示す。ブロック減色処理部 2 0 0 1 は、まず、 $2 \times 2$  画素のブロックを単位に画素値を調査し

50

、 $2 \times 2$ 画素の中で1組でも近い色の画素があれば統合する。この処理を画像全体に対して行い、続いて、補間データ生成部1101で補間データ生成を行う。補間データ生成部1101の処理内容は、前記第3の実施形態で説明したので省略する。ブロック減色処理部2001は、例えば以下のような方法で行う。

【0192】

着目ブロック内の $2 \times 2$ 画素の色の差分をそれぞれ調査し、次のF1乃至F6を算出する。

$$F1 = |X - X_a|$$

$$F2 = |X - X_b|$$

$$F3 = |X - X_c|$$

$$F4 = |X_a - X_b|$$

$$F5 = |X_a - X_c|$$

$$F6 = |X_b - X_c|$$

この時、次式(3)が成り立つ場合には、 $2 \times 2$ 画素の色の統合を行う。

$$Prm > \max(F1, F2, F3, F4, F5, F6) \dots (3)$$

ここで、 $\max(A1, A2, \dots, An)$ は、値A1, A2, ..., Anの中の最大数を返す関数である。

【0193】

色の統合とは、本実施形態では補間対象画素である $X_a, X_b, X_c$ を縮小画像生成のためにサンプリングされた画素Xの色に修整する(置き換える)ものである。つまり、着目ブロックをフラットブロックに変換するものと言えば分かりやすい。なお、 $Prm1 > Prm2$ であるので、 $Prm1$ が選択された場合には、色の統合される可能性が高く、 $Prm2$ が選択された場合には色の統合される可能性は低くなる。

【0194】

また、式(3)が成立しなかった場合、着目ブロックの全画素(4画素)が互いに近い色ではないことになり、統合処理を行わない。しかしながら、そのブロック中の2画素が全く同じではないものの、互いに近い色である場合、その2つの画素が、その2色の平均値で置き換えても良い。理由は、着目ブロックに含まれる色数は減ることはあっても増えることはないからである。仮に、色数が“4”から“3”になった場合には圧縮率が高くならないが、色数が“1”になった場合に着目ブロックがフラットブロックであることになり、高い圧縮率が期待できる。また、色数が“2”の場合には図16のパターンのいずれかに該当することになるので高い圧縮率が期待できる。

【0195】

また、別な減色の方法としては、以下のようにしても良い。

【0196】

まず、 $2 \times 2$ 画素のうち、色距離の一番大きな2画素を探索し、色距離の離れた2画素をそれぞれのグループの代表色と決める。次いで、それぞれの代表色と、代表色でない画素の色距離を求め、色距離の小さい方へグルーピングを行う。グルーピングされた画素をそれぞれ統合することで、 $2 \times 2$ 画素が2色に減色される。グルーピングされた画素の統合の方法は、そのグループの画素の平均値とする。但し、本実施形態では、グループの中に縮小画素Xを含んでいれば、そのグループの画素値は全てXに置き換えるものとする。

【0197】

以上の減色処理を画像全体の $2 \times 2$ 画素のブロックに対して行う。画像全体のブロックに対して減色処理が終了したら、補間データ生成部へ減色処理後の処理後の画像データを出力し、再符号化を行なわせる。

【0198】

補間データ生成部1101には、入力される画像データが、原画像であるか、減色後の画像であるかの違いのみであり、処理の内容は第3の実施形態で説明したロスレスの補間データ生成と違いはない。

【0199】

10

20

30

40

50

以上の方法により、減色処理を行った後で補間データ生成を行うことで、ブロック中に存在する色数が少なくなるため、ノンフラットブロックの2, 3色処理に入り、3画素分の画素値を出力する可能性が低くなるため、補間データ量を抑制する効果がある。また、ロスレス補間生成時に取得した、画素値出力回数  $i$  に応じて、減色処理でのパラメータを調整することで、画質をなるべく保ちつつ、補間データ量を削減することが可能となる。具体的には、ロスレス補間生成時に画素値出力回数  $i$  が多ければ、パラメータを大きくして、色を統合させ易くして、補間データ量の抑制効果を大きくする。一方、画素値出力回数  $i$  が少なければ、それほどパラメータを大きくしないで、色の変化を少なくすることで、画質を保ちつつデータ量を抑制することが可能となる。

#### 【0200】

尚、本実施形態では、画素値出力回数のみをカウントして利用したが、同じ画素値出力回数でも、ブロックが2色の場合には1画素分の画素値を出力し、ブロックが3, 4色の場合には3画素分の画素値を出力するので、補間データ量は変わってくる。そのため、2色処理で画素値を出力したのか(1画素分出力)、3, 4色処理で画素値を出力したのか(3画素分出力)で別々にカウントした情報を保持しておいて、パラメータの設定方法を変更して構わない。

#### 【0201】

ロッシー補間生成(ステップS1907)後、処理をステップS1903へ戻し、ロッシーの補間データ量と予め定められた目標値との比較を行う。この時ロッシーの補間データ量が目標値よりも小さければ(N)、ステップS1908処理を進め、補間データを出力して処理を終了する。一方、目標値よりも大きければ(Y)、パラメータを再設定し、ロッシー符号化を行う。再設定するパラメータは、徐々に大きな値になっていく。なお、最悪の場合、画像中全てのブロックが同色になるが、繰り返し回数を制限して、任意の繰り返し回数を超えた場合にはパラメータを画素値の最大値より大きい値を設定することで、処理を終了させることが可能である。

#### 【0202】

本実施形態におけるロッシー補間生成は、説明を簡易にするため、まず画像全面に対して、ブロックの減色処理を済ませた後、ロスレス補間生成と同じ方法で補間データの生成を行うと説明した。しかし、ロッシー補間生成の方法はこれに限らず、注目しているブロックの減色に続いて、注目ブロックの補間データ生成を行う。このとき、縮小画素としてブロック内の右下画素を抽出することで、周囲画素を参照する場合にも、減色処理後の画素値を参照することができるので、ブロック単位での処理が可能となる。

#### 【0203】

##### [第8の実施形態]

上記の第7の実施形態では、ロスレス補間生成で補間データ量が目標値よりも大きくなってしまった場合に、ロスレス補間生成時に取得した情報によって、ロッシー補間生成で再符号化を行う際のパラメータを決定した。本第8の実施形態では、符号量制御の方法として、ロスレス補間生成で取得した情報により、データの削減方法自体を変更する方法について説明する。

#### 【0204】

図21は、本第8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と同じ処理の内容については説明を省略する。

#### 【0205】

本第8の実施形態では、第7の実施形態と同様に、ロスレスの補間データのデータ量の大小によって、再符号化するか否かを判定する(ステップS1903)。再符号化すると判定された場合に、ステップS1904では、前記第7の実施形態と同様の判定を行うが、本第8の実施形態ではロッシー補間生成時のパラメータ設定値を変更するのではなく、ロッシーの処理方法を切り替える。具体的には、画素値出力回数  $i$  が閾値  $Th3$  よりも大きい場合(Y)には、ステップS2101へ処理を進め、画素値出力回数  $i$  が閾値  $Th3$  以下の場合(N)には、処理をステップS1907へ進め、ロッシー補間生成を行

10

20

30

40

50

う。ロッシー補間生成を行った場合には、処理をステップS 1 9 0 3へ戻して、ロッシーの補間データ量が目標値以下であるか否かの判定を行う。ここで、ロッシー補間のデータ量が、目標値よりも大きいようであれば、処理をステップS 1 9 0 4へ進めるが、本実施形態では、2度目にステップS 1 9 0 4の判定を行う場合には、閾値Th 3に「0」を設定し、必ずステップS 2 1 0 1の処理に移るようにする。

#### 【0206】

ステップS 2 1 0 1は、入力画像データ全体を解像度変換する。ここで、解像度変換の方法は、 $2 \times 2$ 画素を注目している $2 \times 2$ 画素ブロックの平均値に置き換えるものとする。2画素の平均値を取る理由は、画像の劣化がなるべく目立たないようにするためであって、その方法に限らない。解像度変換後のデータ量は、入力画像データの $1/4$ になる。解像度変換を行った場合には、処理をステップS 1 9 0 8へ進め、解像度変換後のデータを出力する。

10

#### 【0207】

以上の処理により、補間データ量がある値以上だった場合には、ロッシー補間生成または、解像度変換処理によって、データ量を削減できる。ロッシー補間生成を行うか、解像度変換処理を行うかを、ロスレス補間生成時の情報を利用することによって、画素値出力回数が非常に多い場合には、直接解像度変換をすることができ、繰り返し回数が少なくなり、処理時間が短縮できる。また、画素値出力回数が非常に多い画像は、画素値の変化が多い高解像度自然画のような画像である可能性が高いため、解像度変換をして解像度を落としても画質の劣化が目立ちにくいいため、画質を保ちつつ、データ量の抑制が可能である。

20

#### 【0208】

以上本発明に係る各実施形態を説明した。本実施形態は、各実施形態の説明で参照した各フローチャートに従った処理に相当するプログラムをコンピュータに実行させることでも実現できる。たとえば、上記実施形態における符号化処理にかかるプログラム、或いは、復号(画像展開)処理にかかるプログラムを、パーソナルコンピュータ(PC)上で実行しても構わない。また、通常、コンピュータプログラムは、CD-ROM等のコンピュータ可読記憶媒体に格納されており、それをコンピュータが有する読取り装置(CD-ROMドライブ等)にセットし、システムにコピーもしくはインストールすることで実行可能になる。従って、かかるコンピュータプログラムも本発明の範疇に入ることは明らかである。

30

#### 【図面の簡単な説明】

#### 【0209】

【図1】第1の実施形態に係る補間データ生成の処理手順を示すフローチャートである。

【図2】第1の実施形態に係る画像処理装置のブロック構成図である。

【図3】実施形態における画像データと $2 \times 2$ 画素のブロックデータとの関係を示す図である。

【図4】着目ブロックデータ内の $2 \times 2$ 画素の配列を示す図である。

【図5】第1の実施形態における第1、第2、第3の付加情報の符号語の例を示す図である。

40

【図6】処理対象画像の例を示す図である。

【図7】第1の実施形態において生成する補間データのデータ構造の例を示す図である。

【図8】第2の実施形態に係る画像処理装置のブロック構成図である。

【図9】第2の実施形態に係る付加情報統合部の処理手順を示すフローチャートである。

【図10】第2の実施形態に係る付加情報統合部の処理例を示すための画像例とその処理結果の例を示すための図である。

【図11】第3の実施形態に係る画像処理装置のブロック構成図である。

【図12】第3の実施形態に係る補間データ生成部のブロック構成図である。

【図13】第3の実施形態に係るフラット判定部の処理手順を示すフローチャートである。

50



【図14】実施形態の画像処理装置をコンピュータによって実現する場合のコンピュータのブロック構成図である。

【図15】第3の実施形態に係るノンフラットブロック解析部の処理手順を示すフローチャートである。

【図16】第3の実施形態に係る2色の配置パターンの種類を示す図である。

【図17】第4の実施形態に係るノンフラットブロック解析部の処理手順を示すフローチャートである。

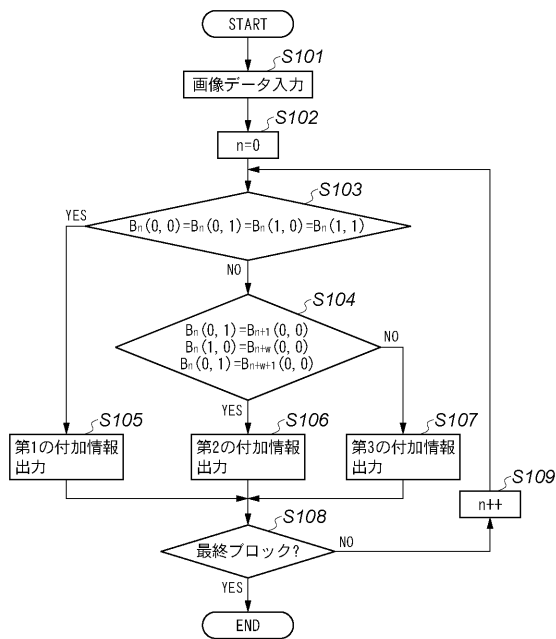
【図18】第6の実施形態に係る着目ブロックとその周囲の画素位置を示す図である。

【図19】第7の実施形態に係る補間データの符号量制御方法の処理手順を示すフローチャートである。

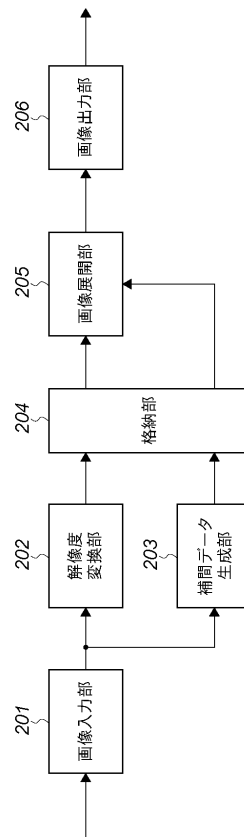
【図20】第7の実施形態に係るロッシェ補間生成のブロック構成図である。

【図21】第8の実施形態に係る補間データの符号量制御方法の処理手順を示すフローチャートである。

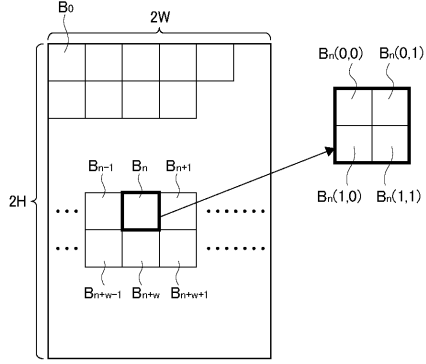
【図1】



【図2】



【図3】



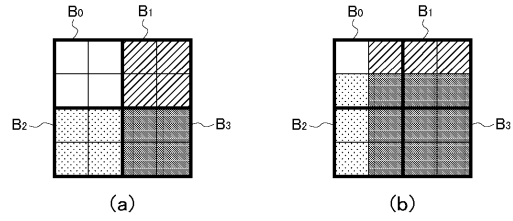
【図4】

X	Xa
Xb	Xc

【図5】

	フラグ	画素値情報
第1の付加情報	1	-
第2の付加情報	01	-
第3の付加情報	00	24bit

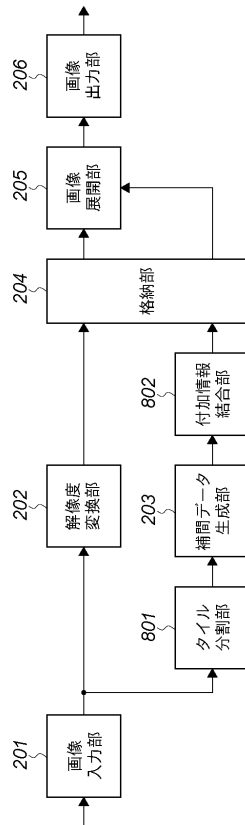
【図6】



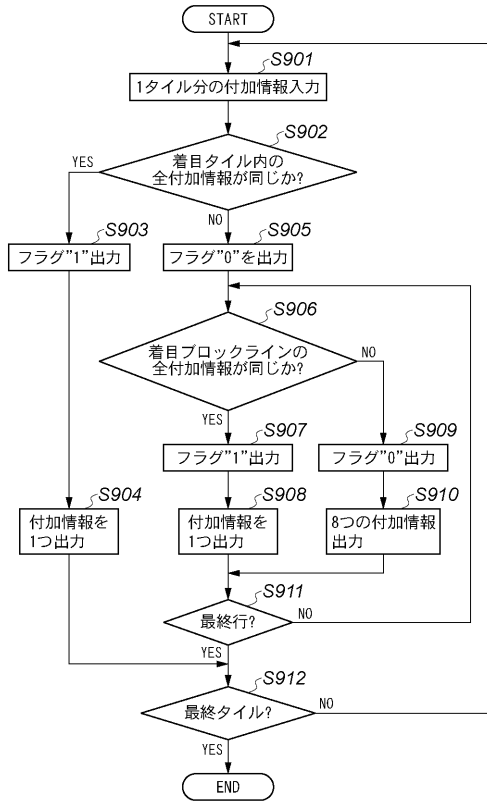
【図7】

01	B <sub>n</sub> (0,1)の画素値	B <sub>n</sub> (1,1)の画素値	00	B <sub>n</sub> (1,0)の画素値	B <sub>n</sub> (0,1)の画素値	B <sub>n</sub> (1,0)の画素値	00	B <sub>n</sub> (1,1)の画素値	B <sub>n</sub> (1,0)の画素値	B <sub>n</sub> (1,1)の画素値	1
----	--------------------------	--------------------------	----	--------------------------	--------------------------	--------------------------	----	--------------------------	--------------------------	--------------------------	---

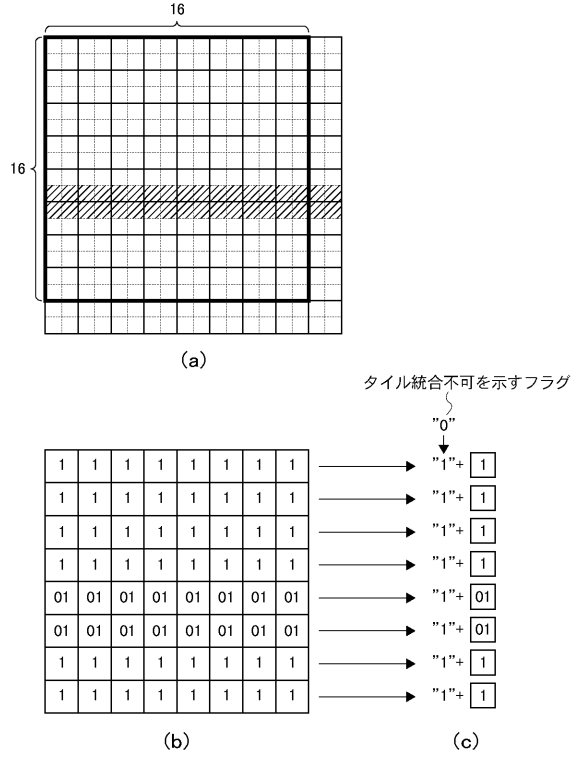
【図8】



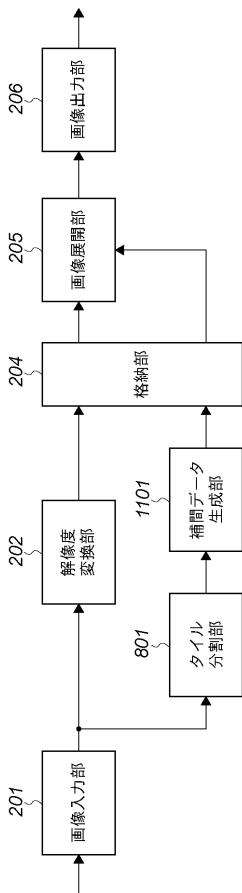
【図9】



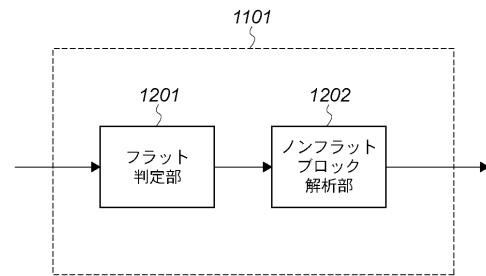
【図10】



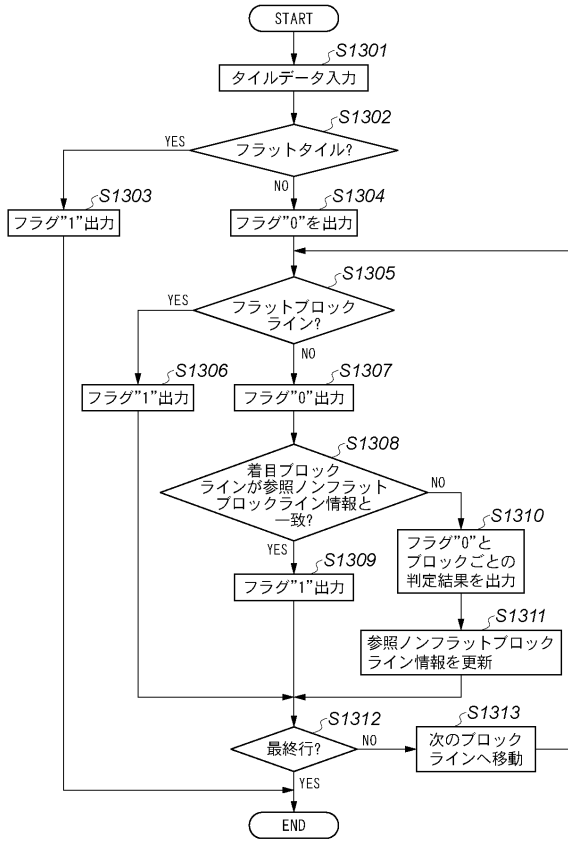
【図11】



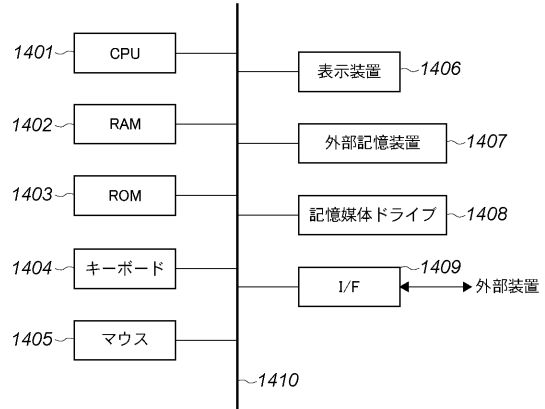
【図12】



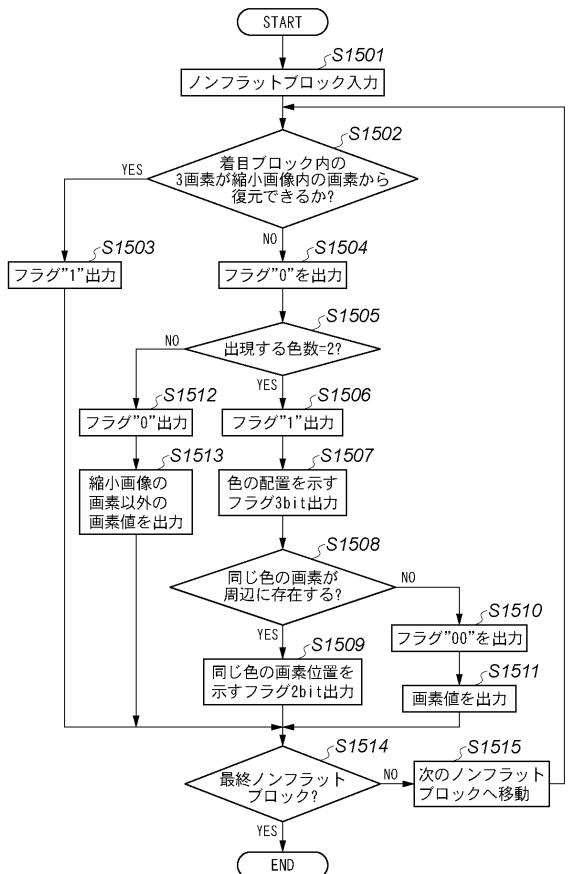
【図13】



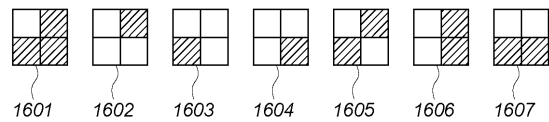
【図14】



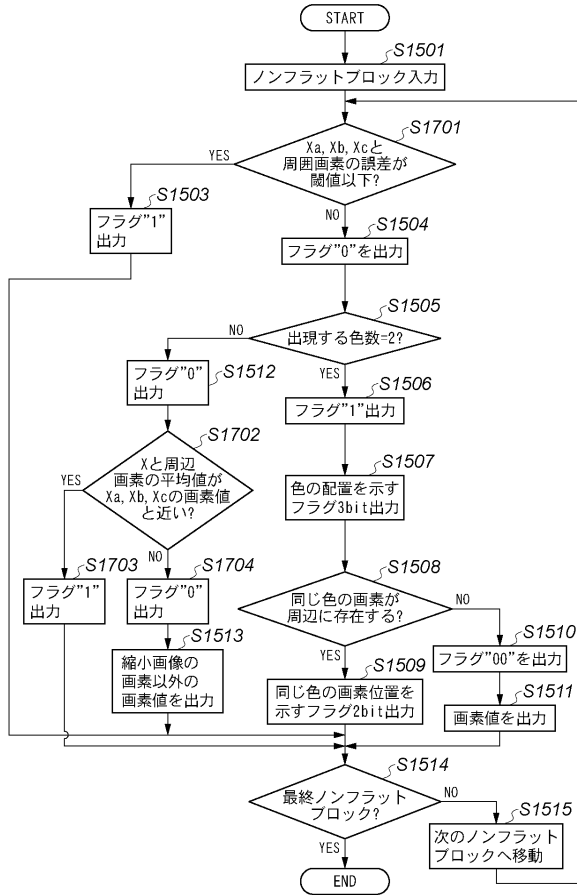
【図15】



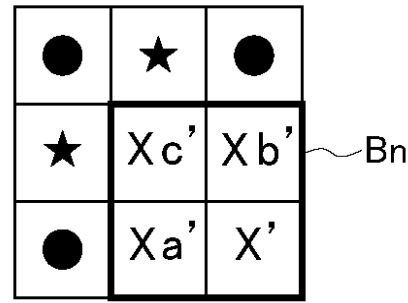
【図16】



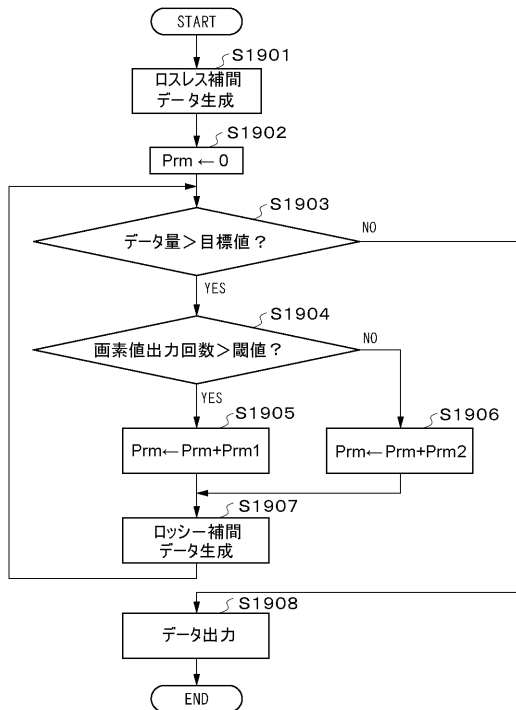
【図17】



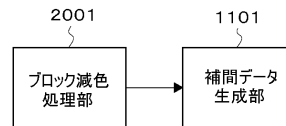
【図18】



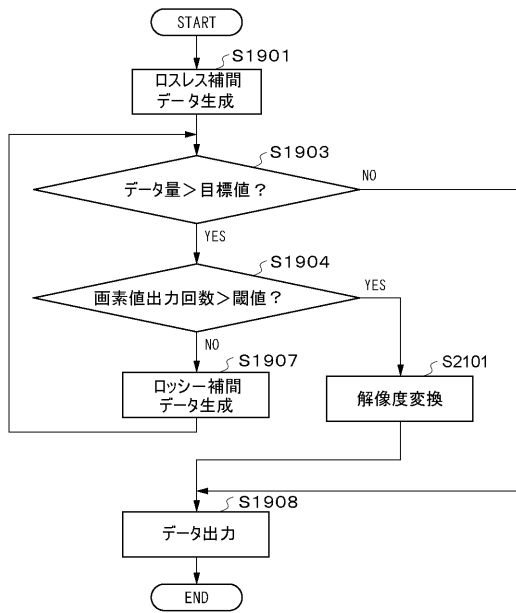
【図19】



【図20】



【図 21】



---

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
- (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 梶原 浩  
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 田村 宏和  
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 隆志

- (56)参考文献 特開平02-211785(JP,A)  
特開平01-183963(JP,A)  
特開2006-340300(JP,A)  
特開平08-051622(JP,A)

(58)調査した分野(Int.Cl., DB名)

H04N 1/41  
H04N 7/26