

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4761566号
(P4761566)

(45) 発行日 平成23年8月31日(2011.8.31)

(24) 登録日 平成23年6月17日(2011.6.17)

(51) Int.Cl.		F I			
HO4N	1/41	(2006.01)	HO4N	1/41	C
HO4N	1/46	(2006.01)	HO4N	1/46	Z
HO4N	1/60	(2006.01)	HO4N	1/40	D

請求項の数 9 (全 29 頁)

(21) 出願番号	特願2006-336384 (P2006-336384)	(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
(22) 出願日	平成18年12月13日(2006.12.13)	(74) 代理人	100076428 弁理士 大塚 康德
(65) 公開番号	特開2008-148264 (P2008-148264A)	(74) 代理人	100112508 弁理士 高柳 司郎
(43) 公開日	平成20年6月26日(2008.6.26)	(74) 代理人	100115071 弁理士 大塚 康弘
審査請求日	平成21年12月14日(2009.12.14)	(74) 代理人	100116894 弁理士 木村 秀二
		(72) 発明者	今福 和也 東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

最終頁に続く

(54) 【発明の名称】 画像処理装置及びその方法とプログラム及び媒体

(57) 【特許請求の範囲】

【請求項1】

画像を複数のブロックに分割し、各ブロックに対してDCTを適用する圧縮方法により生成された圧縮画像を入力して処理する画像処理装置であって、

前記圧縮画像を復号する復号手段による復号過程で、前記ブロックにおけるDCT係数の分布から当該ブロックを分類するブロック分類手段と、

前記復号手段により復号された復号データと前記ブロック分類手段による分類結果から

(1) 直流成分のみからなるブロックについては前記ブロックの1画素の色分解データを生成して当該ブロックの全画素の色分解データとして出力し、

(2) 直流成分及び水平方向の周波数成分からなるブロックについては前記ブロックの1行の画素の色分解データを生成し、当該生成された色分解データを前記ブロックの列方向に繰り返し出力し、

(3) 前記直流成分及び垂直方向の周波数成分からなるブロックについては、前記ブロックの1列の画素の色分解データを生成し、当該生成された色分解データを前記ブロックの行方向に繰り返し出力することで、前記ブロック単位の色分解データを生成する色分解データ生成手段と、

を有することを特徴とする画像処理装置。

【請求項2】

前記色分解データ生成手段は、前記直流成分のみからなるブロックが連続している場合

、直前のブロックの色分解データを採用することを特徴とする請求項1に記載の画像処理装置。

【請求項3】

前記色分解データを量子化して印刷データを生成する量子化手段を更に有することを特徴とする請求項1に記載の画像処理装置。

【請求項4】

前記圧縮画像はJ P E G画像であり、前記ブロックはM C Uであることを特徴とする請求項1乃至3のいずれか1項に記載の画像処理装置。

【請求項5】

前記復号手段は、
ハフマン復号化部と、
逆量子化部と、
逆D C T部とを有することを特徴とする請求項1に記載の画像処理装置。

10

【請求項6】

前記色分解データ生成手段は、
前記復号データを予め定められた色空間へ変換する色空間変換手段と、
前記色空間変換手段により生成される色空間変換データをデバイスの固有色に分解する色分解手段とを有することを特徴とする請求項1に記載の画像処理装置。

【請求項7】

前記色分解データ生成手段は、更に、前記色分解された色分解データを変倍する変倍手段を有することを特徴とする請求項1に記載の画像処理装置。

20

【請求項8】

画像を複数のブロックに分割し、各ブロックに対してD C Tを適用する圧縮方法により生成された圧縮画像を入力して処理する画像処理方法であって、

前記圧縮画像を復号する復号工程での復号過程で、前記ブロックにおけるD C T係数の分布から当該ブロックを分類するブロック分類工程と、

前記復号工程で復号された復号データと前記ブロック分類工程での分類結果から、
(1) 直流成分のみからなるブロックについては前記ブロックの1画素の色分解データを生成して当該ブロックの全画素の色分解データとして出力し、

(2) 直流成分及び水平方向の周波数成分からなるブロックについては前記ブロックの1行の画素の色分解データを生成し、当該生成された色分解データを前記ブロックの列方向に繰り返し出力し、

30

(3) 前記直流成分及び垂直方向の周波数成分からなるブロックについては、前記ブロックの1列の画素の色分解データを生成し、当該生成された色分解データを前記ブロックの行方向に繰り返し出力することで、前記ブロック単位の色分解データを生成する色分解データ生成工程と、

を有することを特徴とする画像処理方法。

【請求項9】

コンピュータを、画像を複数のブロックに分割し、各ブロックに対してD C Tを適用する圧縮方法により生成された圧縮画像を入力して処理する画像処理装置として機能させるためのプログラムであって、該コンピュータを、

40

前記圧縮画像を復号する復号手段による復号過程で、前記ブロックにおけるD C T係数の分布から当該ブロックを分類するブロック分類手段と、

前記復号手段により復号された復号データと前記ブロック分類手段による分類結果から

(1) 直流成分のみからなるブロックについては前記ブロックの1画素の色分解データを生成して当該ブロックの全画素の色分解データとして出力し、

(2) 直流成分及び水平方向の周波数成分からなるブロックについては前記ブロックの1行の画素の色分解データを生成し、当該生成された色分解データを前記ブロックの列方向に繰り返し出力し、

50

(3) 前記直流成分及び垂直方向の周波数成分からなるブロックについては、前記ブロックの1列の画素の色分解データを生成し、当該生成された色分解データを前記ブロックの行方向に繰り返し出力することで、前記ブロック単位の色分解データを生成する色分解データ生成手段とを有する画像処理装置として機能させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、DCTに基づいて圧縮された画像データを処理する画像処理装置及びその方法に関するものである。

【背景技術】

10

【0002】

近年、デジタルカメラが広く普及し、デジタルカメラで撮影された画像データをプリンタで印刷する機会も増えてきている。それに伴い、デジタルカメラを直接プリンタに接続して印刷するカメラダイレクト印刷や、デジタルカメラのメモリカードをプリンタに装着して、そこに記憶されている画像を印刷するカードダイレクト印刷等が出現している。これら技術では、プリンタがデジタルカメラ或はメモリカードから画像データを取り込み、その画像データをデコードして印刷データを生成して印刷している。

【0003】

このような画像データを処理するプリンタのリソースは、PCが有するリソースに比べて貧弱であるため、ダイレクト印刷を行うシステムでは、特に印刷データを生成するための処理を効率的に行うことが望まれている。

20

【0004】

このような背景がある一方で、デジタルカメラの進歩は加速度的であり、デジタルカメラで撮影可能な画素数も急激に増加している。また画質を極力落とさず画像データの容量を抑える目的で、画像データをメモリカードに保存する際にDCTに基づいたカラー画像の圧縮フォーマット、例えばJPEGが標準的なフォーマットとして扱われている。このJPEGでは、画像データをMCU(Minimum Coded Unit)と呼ばれる複数のブロックに分割し、各MCUをDCT変換している。そして、高周波信号に対する感度が低いという人間の視覚特性を利用して、そのDCT係数の高周波成分を削ることで高圧縮を実現している。しかし、このようなJPEGフォーマットへの符号化、復号化処理にかかる負荷は

30

【0005】

そこで特許文献1は、JPEGの特性を利用した処理の効率化技術として、MCUの周波数特性を考慮し、ベースライン方式のJPEG圧縮処理を高速化する技術を提案している。その内容は、画像データをMCUに分割して、各MCUを単一色、非単一色に分類し、単一色は簡単な計算でDC係数のみを求めて符号化し、非単一色は通常の符号化を行うといったものである。このようにJPEGで扱われるDCTの特性に着目して符号化及び復号化処理を軽減するための技術も提案されているが、あくまで単一色からなるMCUを効率的に扱う内容に留まっている。

【0006】

40

またプリンタでは、JPEG画像等のデコード以外にも、デコードしたカラー画像データから、プリンタが印刷可能な印刷データを生成しなければならない。このデコード後のカラー画像データから効率的に印刷データを生成するための技術について次に述べる。

【0007】

一般的にカラー画像データはプリンタ固有の色空間とは異なる、例えばCRT等の色空間で表現される。カラー画像データが表現する色空間には多種多様なものが存在するため、一度標準的な色空間に変換した後、プリンタ固有の色材の色に分解する必要がある。このようにプリンタ固有の色材の色データを生成するためには膨大な処理が必要となるため、これらの処理を効率的に行うための技術が提案されている。

【0008】

50

特許文献2では、色変換処理、色補正、色変換を行う印刷処理において、色空間変換による処理速度の低下を改善するための技術が提案されている。その内容は、入力データとデバイス依存データを関連づけて保存し、色処理キャッシュとして利用するというものである。キャッシュ領域は複数のキャッシュデータを保持できる構成になっており、入力データとその入力データより得られる色処理結果の組み合わせが既にキャッシュされているかを、キャッシュデータテーブルを検索して判定している。ここでキャッシュデータテーブルにキャッシュされていた場合は、そのキャッシュされているデータを利用することで色処理を省略する。またキャッシュデータテーブルにキャッシュされていなかった場合は、その入力データに対して色処理を施し、その結果をキャッシュ領域の空き領域、もしくは既に使用されている領域に上書きする。このような処理により、色処理にかかる処理量を削減している。とりわけ、保持するキャッシュデータを増やすほど、一般的にキャッシュのヒット率が上がる。

10

【0009】

一方、この技術には、次のような短所もある。複数のキャッシュデータをテーブル形式で保持しなければならないため、テーブルサイズを大きくするほどキャッシュ領域にかかるコストが増加してしまう。また、テーブルサイズを大きくするほどキャッシュデータテーブルの検索に時間がかかってしまう。

【特許文献1】特開2001-145107号公報

【特許文献2】特開2005-53009号公報

【発明の開示】

20

【発明が解決しようとする課題】

【0010】

上述のようにプリンタ本体が頻繁に扱う画像フォーマット固有の処理負荷を軽減するための技術や、画像データから印刷データを生成する際に行う色処理の効率化に関する技術がこれまで提案されてきているが、改善の余地はまだ存在する。

【0011】

本発明の目的は、上述した問題点を解決することにある。

【0012】

本願発明の特徴は、DCTに基づき圧縮されたカラー画像データに対する色処理の更なる高速化を実現することを目的とする。

30

【課題を解決するための手段】

【0013】

上記目的を達成するために本発明の一態様に係る画像処理装置は以下のような構成を備える。即ち、

画像を複数のブロックに分割し、各ブロックに対してDCTを適用する圧縮方法により生成された圧縮画像を入力して処理する画像処理装置であって、

前記圧縮画像を復号する復号手段による復号過程で、前記ブロックにおけるDCT係数の分布から当該ブロックを分類するブロック分類手段と、

前記復号手段により復号された復号データと前記ブロック分類手段による分類結果から

40

(1) 直流成分のみからなるブロックについては前記ブロックの1画素の色分解データを生成して当該ブロックの全画素の色分解データとして出力し、

(2) 直流成分及び水平方向の周波数成分からなるブロックについては前記ブロックの1行の画素の色分解データを生成し、当該生成された色分解データを前記ブロックの列方向に繰り返し出力し、

(3) 前記直流成分及び垂直方向の周波数成分からなるブロックについては、前記ブロックの1列の画素の色分解データを生成し、当該生成された色分解データを前記ブロックの行方向に繰り返し出力することで、前記ブロック単位の色分解データを生成する色分解データ生成手段と、を有することを特徴とする。

【0014】

50

上記目的を達成するために本発明の一態様に係る画像処理方法は以下のような工程を備える。即ち、

画像を複数のブロックに分割し、各ブロックに対してDCTを適用する圧縮方法により生成された圧縮画像を入力して処理する画像処理方法であって、

前記圧縮画像を復号する復号工程での復号過程で、前記ブロックにおけるDCT係数の分布から当該ブロックを分類するブロック分類工程と、

前記復号工程で復号された復号データと前記ブロック分類工程での分類結果から、

(1) 直流成分のみからなるブロックについては前記ブロックの1画素の色分解データを生成して当該ブロックの全画素の色分解データとして出力し、

(2) 直流成分及び水平方向の周波数成分からなるブロックについては前記ブロックの1行の画素の色分解データを生成し、当該生成された色分解データを前記ブロックの列方向に繰り返し出力し、

(3) 前記直流成分及び垂直方向の周波数成分からなるブロックについては、前記ブロックの1列の画素の色分解データを生成し、当該生成された色分解データを前記ブロックの行方向に繰り返し出力することで、前記ブロック単位の色分解データを生成する色分解データ生成工程と、を有することを特徴とする。

【発明の効果】

【0015】

本発明によれば、DCTの周波数特性を利用してMCUを分類することにより画像データのパターンを特定するため、キャッシュデータテーブルの検索が不要となり効率的な処理を実現できる。

【発明を実施するための最良の形態】

【0016】

以下、添付図面を参照して本発明の好適な実施の形態を詳しく説明する。尚、以下の実施の形態は特許請求の範囲に係る本発明を限定するものでなく、また本実施の形態で説明されている特徴の組み合わせの全てが本発明の解決手段に必須のものとは限らない。尚、以下の説明では便宜上、プリンタが受信した画像データに基づいて、そのプリンタで印刷データを生成してプリントする機能を「ダイレクトプリント」と呼ぶこととする。

【0017】

まず、以下のいくつかの実施形態において共通に用いられる画像処理システムの全体概要、ハードウェア構成の概要、ソフトウェア構成の概要について説明する。

【0018】

図1は、本発明の実施の形態に共通するプリントシステム(画像処理システム)の概略構成を示すブロック図である。

【0019】

このプリントシステムは、PC等で構成されるホスト装置11と、プリンタ等で構成される画像処理装置12とを備え、これらが双方向インターフェース13を介して接続されている。

【0020】

次に、画像処理装置12のハードウェア構成の概要について説明する。

【0021】

図2は、本実施の形態に係るプリントシステムを構成する画像処理装置12のハードウェアの概略構成を説明するブロック図である。

【0022】

図2において、1000は制御基板を示している。ASIC1001は、このASIC1001に接続された各部とのインターフェースを制御するインターフェース部(不図示)を備えるとともに、画像データの生成等の各種処理を実行するCPU1001aも備えている。コネクタ1002は、制御基板1000とオペレーションパネル1003とを接続する。ユーザは、このオペレーションパネル1003を操作して、各種印刷設定情報を設定したり、メンテナンスの指示、メモリカード(MC)1007に記憶されている画像

10

20

30

40

50

の閲覧及び印刷画像の指定、印刷開始指定、印刷のキャンセル等を行うことができる。コネクタ1004は、このオペレーションパネル1003とビューワ1005とを接続している。ビューワ1005は、メモリカード1007に記憶された画像の閲覧用の表示、GUIによるメニュー表示、各種エラーメッセージ等を表示する。コネクタ1006は、このメモリカード1007と制御基板1000とを接続しており、ASIC1001は、このメモリカード1007から画像データを直接読み込んで印刷することができる。

【0023】

DSC用USBバスコネクタ1008は、デジタルカメラ(DSC)1010と制御基板1000とを接続するUSBバスコネクタであり、コネクタ1009は、デジタルカメラ1010を接続するためのコネクタである。デジタルカメラ1010は、カメラ内部のメモリに保存している画像データ及び印刷設定情報を、この画像処理装置12に対して出力可能に構成されている。尚、このデジタルカメラ1010の構成としては、内部に記憶手段としてのメモリを備えるものや、取り外し可能なメモリを装着するためのスロットを備えたもの等、種々の構成を採用することができる。これによりデジタルカメラ1010内の画像データを、そのデジタルカメラ1010で設定された印刷設定情報に従って処理し、カメラダイレクトプリントを行うことができる。

10

【0024】

USBバスハブ1011は、接続コネクタ1012を経由して接続されたPC1013からの画像データを印刷する際には、PC1013からのデータをASIC1001に渡す。メモリ1014は、ASIC1001のCPU1001aの制御プログラム等を記憶するプログラムメモリを有している。更に、実行時のプログラムを記憶するRAMエリア、更には画像データや印刷設定情報、後述する各種フラグや変数等を記憶するワークメモリエリアを有している。

20

【0025】

コネクタ1015は、CR(キャリッジ)制御基板及びプリントヘッド1016と制御基板1000とを接続している。1017はLF(用紙送り)モータドライバ、1018はLFモータ1019を接続するためのコネクタである。同様に、1020はASF(自動用紙給送)モータドライバ、1021はASFモータ1022を接続するためのコネクタである。1023は電源用コネクタで、電源1024により直流電圧を入力している。この電源1024は、電池或は商用電源からDC電圧を生成するACアダプタであっても良い。

30

【0026】

次に本実施の形態に係る画像処理装置12のソフトウェア構成の概要について説明する。ここではプリント機能について説明する。

【0027】

図3は、本実施の形態に係る画像処理装置12のプリント機能で用いられるソフトウェアのモジュール構造を示す図である。以下図3を参照して、プリント処理を説明する。尚、ここでは、メモリカード1007から画像データを読み込んでプリントする場合、PC1013からデータを受信してプリントする場合、デジタルカメラ1010から画像データを受信してプリントする場合を説明する。尚、この図3に示す各モジュールやラスタライザ、インターフェース制御モジュールは、メモリ1014に記憶されてプログラムとCPU1001aにより実現されている。

40

【0028】

まず初めに、メモリカード1007から画像データを入力してプリントする場合を説明する。

【0029】

ダイレクトプリント機能制御モジュール2001は、DSC用USB I/Fモジュール2002やメモリカードI/F2004、PC用USB I/Fモジュール2017と連携して、DSC1010、メモリカード1007、PC1013との接続状態を把握する。その接続状態は、随時オペレーションパネルI/Fモジュール2006に通達され、その

50

状態がオペレーションパネルに1003に表示される。即ち、現在メモ리카ード1007、デジタルカメラ1010との間でダイレクトプリントが可能かどうかを判断し、ユーザがオペレーションパネル1003で指定できるユーザ設定メニューを表示する。また、オペレーションパネルI/F2006は、DSC1010やメモ리카ード1007との接続状態に拘わらず、常に用紙設定等の各種印刷に関する設定を受け付け、現在の印刷設定情報を記憶しておく。これら印刷設定方法を指定するためのメニュー画面等は、ビューワ1005に表示される。オペレーションパネル用I/Fモジュール2006とビューワ用I/Fモジュール2008は連携して、オペレーションパネルのキー操作情報を共有する。またDSC1010やメモ리카ード1007、PC1013の接続状態情報も共有する。そしてそれらの共有情報に基づきビューワ1005に設定可能なメニュー画面や接続されたデバイス内の画像データの表示等を行う。

10

【0030】

メモ리카ード1007が接続されている場合、ユーザはオペレーションパネル1003とビューワ1005とを利用して、メモ리카ード1007の画像を閲覧したり、印刷する画像の指定、印刷開始の指示を行える。ユーザがメモ리카ード1007の画像データを印刷する場合、オペレーションパネル1003の印刷開始キーが押されると、オペレーションパネル用I/Fモジュール2006からダイレクトプリント機能制御モジュール2001に対して印刷開始要求が発行される。この印刷開始要求を受けたダイレクトプリント機能制御モジュール2001は、それをメイン制御モジュール2000に通知する。メイン制御モジュール2000は、メモ리카ード1007以外のDSC1010やPC1013からのデータを受け付けなくするようにDSC用USB I/Fモジュール2002やPC用のUSB I/Fモジュール2017に指示する。これらの指示が伝わり、メモ리카ード1007からの画像データの印刷に専念できる状態になった後、メイン制御モジュール2000はダイレクトプリント機能制御モジュール2001にダイレクトプリント処理に遷移する許可を与える。この許可を受けた後、ダイレクトプリント機能制御モジュール2001はラスタライザマネージャ2010に印刷指示を与える。

20

【0031】

ラスタライザマネージャ2010は、現在のプリントモードにあったラスタライザモジュールを選択してラスタ処理を実行させる。例えば、予め生成された印刷データをPC1013から受信して印刷する場合は、PC1013から受信するデータは印刷データそのものである。それに対し、DSC1010やメモ리카ード1007からの画像データを印刷する場合は、ラスタライザモジュール2013が印刷データを生成する必要がある。そのため、予め生成された印刷データをPC1013から受信して印刷する場合は、ラスタライザマネージャ2010は非ダイレクトプリント用ラスタライザモジュール2011を選択してPC1013からの印刷データを処理させる。また、DSC1010やメモ리카ード1007から画像データを受信して印刷する場合は、ジョブ生成モジュール2012及びラスタライザモジュール2013に処理させる。

30

【0032】

ジョブ生成モジュール2012は、ラスタライザマネージャ2010から印刷処理の指示を受ける。ジョブ生成モジュール2012は、ダイレクトプリント機能制御モジュール2001に対して、メモ리카ード1007からのダイレクトプリントか、或はデジタルカメラ1010からのダイレクトプリントであるかを問い合わせる。メモ리카ード1007からのダイレクトプリントであると通知された場合、オペレーションパネル用I/Fモジュール2006からユーザが設定した印刷設定情報を取得してジョブを生成する。その後、ジョブ生成モジュール2012は、ラスタライザモジュール2013に印刷処理指示を与える。

40

【0033】

ラスタライザモジュール2013は、そのジョブ情報に基づいて印刷データを生成する。このジョブ情報は、用紙サイズや用紙の種類、1枚割り付け印刷(n-in-1)やインデクス印刷等の各種印刷方法の指示を含む。更に、印刷品位、各種画像補正に関する指定、

50

印刷指定された画像ファイルに関する情報、メモリカード1007、デジタルカメラ1010のいずれからのダイレクト印刷指示であるかを示す情報を含んでいる。

【0034】

ラスタライザモジュール2013は、まずジョブ情報を解釈し、用紙の種類や印刷品位からエンジン部の動作モードを決定してエンジン制御モジュール2015に伝える。次に、用紙サイズや印刷方法に従って、1枚の用紙のどの位置にどのような大きさで画像データを割り付けるかといったレイアウト情報を形成し、印刷指定された画像ファイルに関する情報を用いて印刷データを生成する。データ変換モジュール2014は、ラスタライザモジュール2013や非ダイレクトプリント用ラスタライザモジュール2011により生成された印刷データを、エンジン部が受け付けるデータ形式に変換する。

10

【0035】

図4は、本実施の形態に係るデータ変換モジュール2014による印刷データの生成処理を説明するフローチャートである。尚、この処理を実行するプログラムはメモリ1014に記憶されており、CPU1001aの制御の下に実行される。

【0036】

まずステップS1で、印刷指定された画像ファイルを読み込んで、それをデコードする。次にステップS2に進み、デコード後の画像データの色空間を、この画像処理装置12に固有の色空間へ変換する。次にステップS3に進み、レイアウト情報に従って、その画像データを変倍する。最後にステップS4で、その変倍後の画像データの各画素を量子化(ハーフトニング)する。尚、この量子化は、画像処理装置12が処理するデータの形態に依存する。例えば、画像処理装置12による印刷が2値データに基づいて実行される場合は2値化し、多値データ(濃淡インクによる記録、大小インクによる記録を行うため)に基づいて印刷する場合は多値化する。以上が本実施の形態に係るデータ変換モジュール2014による印刷データの生成処理の概要である。

20

【0037】

更にラスタライザモジュール2013は、必要に応じて給紙や排紙、フィード、プリント等の要求をエンジン制御モジュール2015に対して発行し、エンジン制御モジュール2015とメッセージをやり取りして、同期を取りながら処理を進めていく。メモリカード1007からのダイレクトプリントの場合、印刷指定された画像ファイルへのアクセスはメモリカード用I/Fモジュール2004を通して行う。また、デジタルカメラ1010からのダイレクトプリントの場合は、DSC用USB I/Fモジュール2002を通して行う。

30

【0038】

データ変換モジュール2014は、ラスタライザモジュール2013や非ダイレクトプリント用ラスタライザモジュール2011により生成された印刷データを、この画像処理装置12のエンジン部(不図示)が受け付けるデータ形式に変換する。そして所定量の変換データが揃うごとにメイン制御モジュール2000を経由してエンジン制御モジュール2015に通知する。エンジン制御モジュール2015は、エンジン部の動作モードを考慮して、データ変換モジュール2014で変換されたデータをエンジン部に渡して印刷処理を行わせる。

40

【0039】

次にデジタルカメラ1010から画像データを受信して印刷するダイレクトプリントを行う場合の処理を説明する。

【0040】

ダイレクトプリント機能制御モジュール2001は、メモリカード1007からの画像データを印刷するダイレクトプリントの場合と同様の処理を行う。

【0041】

デジタルカメラ1010が接続されている場合、ユーザはデジタルカメラ1010に備えられた機能を利用してデジタルカメラ1010の記憶部に記憶されている画像の閲覧、印刷する画像の指定、印刷設定、印刷開始指示を行う。デジタルカメラ1010からのダ

50

レクト印刷の場合、デジタルカメラ1010の印刷開始キーが押されると、まずDSC用USB I/Fモジュール2002からダイレクトプリント機能制御モジュール2001に対して印刷開始要求が発行される。この印刷開始要求を受けたダイレクトプリント機能制御モジュール2001は、それをメイン制御モジュール2000に通知する。メイン制御モジュール2000は、デジタルカメラ1010以外のメモ리카ード1007やPC1013からのデータを受け付けなくするようにメモ리카ード用I/Fモジュール2004やPC用のUSB I/Fモジュール2017に指示する。これらの指示が伝わってデジタルカメラ1010からの印刷に専念できる状態になった後、メイン制御モジュール2000は、ダイレクトプリント機能制御モジュール2001にダイレクトプリント処理に遷移する許可を与える。

10

【0042】

この場合のダイレクトプリント機能制御モジュール2001の動作は、メモ리카ード1007からのダイレクトプリントを行う場合と同様である。

【0043】

ジョブ生成モジュール2012は、ラスタライザマネージャ2010から印刷処理指示を受ける。ジョブ生成モジュール2012は、ダイレクトプリント機能制御モジュール2001に対して、メモ리카ード1007からのダイレクトプリントか、デジタルカメラ1010からのダイレクトプリントであるかを問い合わせる。その結果、デジタルカメラ1010からのダイレクトプリントであると通知された場合、DSC用USB I/Fモジュール2002から、ユーザが設定した印刷設定情報を取得してジョブを生成する。その後、ジョブ生成モジュール2012は、ラスタライザモジュール2013に印刷処理指示を与える。このときのラスタライザモジュール2013の動作は、メモ리카ード1007からのダイレクトプリントを行う場合とほぼ同様である。ただ異なる点は、DSC用USB I/Fモジュール2002経由で所望の画像ファイルにアクセスする点にある。また、データ変換モジュール2014とエンジン制御モジュール2015による処理は、メモ리카ード1007からのダイレクトプリントを行う場合と同様の動作がなされる。以上がソフトウェア構成の概要説明である。

20

【0044】

本実施の形態の特徴は、図3のラスタライザモジュール2013で実施される印刷データの生成処理に関する。特にDCTに基づき圧縮されたカラー画像データをデコードし、画像データの色空間をデバイス固有の色空間へ変換する処理に関するものである。

30

【0045】

以下、DCTに基づいて圧縮されたカラー画像データに対する印刷データの生成処理に着目して実施の形態1について説明する。また説明の便宜上、DCTに基づいて圧縮されたカラー画像データとしてJPEGを例に説明する。

【0046】

[実施の形態1]

本実施の形態1では、JPEGデータからデバイス固有の色空間に変換された色分解データを高速に生成することを目的としている。

【0047】

前述の図4で説明したように、デバイス固有の色空間に変換された色分解データは次のようなステップで生成される。

40

(1) 印刷指定された画像ファイルを読み込んで、その画像データをデコードする。

(2) デコード後の画像データの色空間をデバイス固有の色空間へと変換する。

【0048】

図5は、本実施の形態に係る色空間の変換処理を説明する図である。この処理は図5のフローチャートのステップS1, S2の処理に相当している。

【0049】

まずメモ리카ード1007やデジタルカメラ1010、PC1013から受信したJPEGデータは、ハフマン復号化部500により、DC成分、AC成分ともにハフマン展開

50

される。これにより量子化されたDCT係数データがMCU毎に得られる。次にハフマン復号化部500から出力された量子化されたDCT係数データは、逆量子化部501で逆量子化され、各MCUのDCT係数データが得られる。次に各MCUのDCT係数データは、各MCU毎に逆DCT(IDCT)部502で逆DCT変換され、JPEGデコードデータが出力される。このJPEGデコードデータの色空間は特に定められていない。しかしカメラファイルシステム規格、DCF(Design rule for Camera File system) Version 1.0では、データ形式をYCbCrにするように明記されている。これを踏まえ、ここではJPEGデコードデータとしてYCbCrデータが出力されることとする。

【0050】

また逆量子化部501で生成された各MCUのDCT係数データは同時に、MCU分類情報生成部503に入力される。MCU分類情報生成部503は、DCT係数データを解析し、各MCUを4つのカテゴリに分類し、その分類情報を出力する。

【0051】

以下、図6(A)~(E)、図7(A)~(D)を参照して、その分類方法を説明する。

【0052】

図6(A)~(E)は、本実施の形態に係る逆量子化部501により得られるDCT係数を説明する図で、図中のm, nは、それぞれDCT係数ブロックにおける水平方向の周波数成分(m=0~7)と垂直方向の周波数成分(n=0~7)を表している。

【0053】

図7(A)~(D)は、IDCT部502により得られる画素値を表しており、図中のx, yは、該当MCU内の画素位置を表している。

【0054】

図6(A)は、逆量子化部501で生成された、あるMCUのDCT係数ブロック600を表している。このMCUは8x8画素で構成されるため、それに対応するDCT係数も8x8個の成分が得られる。

【0055】

下式(1)は、このIDCT部502における変換式を表しており、式中のm, nは、図6(A)~(E)のm, nに相当している。

【0056】

$$S_{xy} = (1/4) \cdot C_m C_n S_{mnc} \cos\{(2x+1)m/16\} \cos\{(2y+1)n/16\} \quad \dots \text{式(1)}$$

$$C_m, C_n = (1/\sqrt{2}) \quad (m, n = 0 \text{ のとき})$$

$$C_m, C_n = 1 \quad (\text{それ以外のとき})$$

但し、ここで式(1)左側の S_{mnc} はm=0~7の和を、右側の $C_m C_n$ はn=0~7の和を、それぞれ示している。また C_m は、水平方向の周波数成分がmの時の変換係数を示し、 C_n は垂直方向の周波数成分がnの時の変換係数を示している。 S_{mnc} は、周波数成分(m, n)での周波数成分値を示し、 S_{xy} は、IDCT部502により得られる該当MCUの位置(x, y)での画素値を表わしている。

【0057】

上記式(1)からも分かる通り、図6(B)の601で示される係数、つまり(m, n)=(0, 0)である係数は、DC成分を表している。また図6(C)の602で示される係数、つまりn=0となる係数群は、DC成分及び水平方向のAC成分で構成されていることがわかる。同様に、図6(D)の603で示される係数、つまりm=0となる係数群は、DC成分及び垂直方向のAC成分で構成されていることもわかる。以上のことから次のことが言える。

【0058】

あるMCUのDCT係数が、図6(B)に示されるDC成分(601)のみを持つ場合、そのMCUは図7(A)に示すような、全ての画素が単一色からなるMCUであることが分かる。尚、以下の説明の便宜上、図7(A)のように全ての画素が単一色からなるデ

10

20

30

40

50

ータパターンを「パターンA」と呼ぶことにする。

【0059】

また、あるMCUのDCT係数が、図6(C)の602で示されるDC成分及び水平方向のAC成分を持つような場合、そのMCUは図7(B)に示すような縦じま模様、つまりy方向の8画素に同じ画素値が連続するMCUであることが分かる。尚、以下の説明の便宜上、図7(B)のようにy方向の8画素に同じ画素値が連続するデータパターンを「パターンB」と呼ぶことにする。

【0060】

更に、あるMCUのDCT係数が、図6(D)の603で示されるDC成分及び垂直方向のAC成分を持つような場合、そのMCUは図7(C)に示すような横じま模様、つまりx方向の8画素に同じ画素値が連続するMCUであることが分かる。尚、以下の説明の便宜上、図7(C)のようにx方向の8画素に同じ画素値が連続するデータパターンを「パターンC」と呼ぶことにする。

【0061】

最後に上記のいずれにも該当しない場合、図7(D)に示すような水平方向と垂直方向の周波数成分が混在した、つまりy方向及びx方向ともにデータの連続性が保証されないMCUであることが分かる。尚、以下の説明の便宜上、このようなx及びy方向ともにデータの連続性が保証されないデータパターンを「パターンD」と呼ぶ。このようにDCT係数を解析することで、MCUの画素値パターンを特定することができる。

【0062】

このようにしてMCU分類情報生成部503は、DCT係数値の分布を基に、そのMCUの画素値パターンを特定付けて、その情報をMCU分類情報520として、色分解データ生成部504に出力する。

【0063】

図6(E)は、このMCU分類情報生成部503で生成されて出力されるMCU分類情報520の具体例を示している。

【0064】

この例では、MCUを、上述の4つ(「パターンA」～「パターンD」)に分類するため、MCU分類情報520は、1MCU当り2ビットである。そして「パターンA」であるMCUに対しては「1」を、「パターンB」であるMCUには「2」を、「パターンC」であるMCUには「3」を、最後に「パターンD」であるMCUに「0」を出力している。以上が、このMCUのカテゴリ分類方法に関する説明である。

【0065】

このようにして生成されたMCU分類情報520と、JPEGデコードデータは色分解データ生成部504に入力される。

【0066】

この色分解データ生成部504は、色空間変換部505と色分解部506とを備えている。色空間変換部505は、入力されたJPEGデコードデータの色空間を色分解部506に入力可能な、予め定められた色空間に変換する。ここでは例えば、予め定められた色空間としてsRGBが設定されている場合、色空間変換部505では、YCbCrのJPEGデコードデータをマトリクス演算等によりsRGBに変換する。この色空間の変換方法については様々な技術が存在するが、本実施の形態では、どの技術を用いるかは特に規定しない。

【0067】

色分解部506は、入力された画像データを、LUT等を用いた更なる変換処理によってデバイス固有の色空間へと変換し、更に必要に応じて階調補正も施した後、出力する。例えば、予め定められた色空間としてsRGBが設定されており、またデバイス固有の色空間としてCMYK色空間が設定されている場合、sRGBデータが入力されると、それを変換したCMYKデータ(色分解データ)が出力される。尚、デバイス固有の色空間としてはCMYK色空間に限定されるものではなく、デバイス固有のRGB色空間であっても良い

10

20

30

40

50

。この場合デバイス固有のRGB色空間を別途CMYK色空間に変換する必要がある。このようなデバイス固有の色空間へと変換する色分解処理についても様々な技術が存在するが、本実施の形態では、どの技術を用いるべきか特に規定しない。

【0068】

色分解データ生成部504は、色空間変換部505と色分解部506とを使用し、入力されたJPEGデコードデータからデバイス固有の色空間データを生成して出力する。但し、この際に入力されたJPEGデコードデータの全画素に対して色空間変換部505及び、色分解部506の処理を行うと演算量やメモリアクセス回数が膨大になってしまう。そこで、MCU分類情報520から特定できる各MCU内のデータパターンを考慮した色処理キャッシュを行うことで、色変換のための演算量と、メモリアクセス回数の大幅な削減を実現する。

10

【0069】

次に図8～図12を参照して、本実施の形態に係る色分解データ生成部504における処理を説明する。

【0070】

図8は、本発明の実施の形態1に係る色分解データ生成部504で実行される処理を説明するフローチャートである。ここでは、色分解データ生成部504に、あるMCUのデコードデータと、そのMCUのMCU分類情報520が入力されたときの処理を示している。尚、この処理を実行するプログラムはメモリ1014に記憶されており、CPU1001aの制御の下に実行される。

20

【0071】

まずステップS11で、入力したMCUのMCU分類情報520が「1」かどうか、即ち、前述した「パターンA」であるか否かを判定する。MCU分類情報520が「1」であればステップS12に進み、この「パターンA」に対応する色分解データ生成処理1を実行する。尚、このステップS12の「色分解データ生成処理1」は、図9のフローチャートを参照して詳しく後述する。

【0072】

ステップS11で、「1」でなかった場合はステップS13に進み、MCU分類情報520が「2」かどうか、即ち、前述した「パターンB」であるか否かを判定する。MCU分類情報520が「2」であればステップS14に進み、この性質を利用した色分解データ生成処理2を実行する。このステップS14の「色分解データ生成処理2」は、図10のフローチャートを参照して詳しく後述する。

30

【0073】

ステップS13で、「2」でなかった場合はステップS15に進み、MCU分類情報520が「3」かどうか、即ち、前述した「パターンC」であるか否かを判定する。MCU分類情報520が「3」であればステップS16に進み、この性質を利用した色分解データ生成処理3を実行する。このステップS16の「色分解データ生成処理3」は、図11のフローチャートを参照して詳しく後述する。

【0074】

ステップS15で、「3」でなかった場合、そのMCUは「パターンD」であることがわかるためステップS17で、この性質を考慮した色分解データ生成処理4を実行する。尚、このステップS17の「色分解データ生成処理4」は、図12のフローチャートを参照して詳しく後述する。

40

【0075】

図8のフローチャートは、色分解データ生成部504において、入力したMCU分類情報520に従って色分解データの生成処理を切り替える処理を説明している。

【0076】

次に図9を参照して「色分解データ生成処理1」について説明する。

【0077】

図9は、図8のステップS12の「パターンA」のMCUに対応する色分解データの生

50

成処理 1 を説明するフローチャートである。この「パターン A」は、MCU 内の 8×8 画素の全てが同じ値である。よって、この処理では、MCU 内の 1 画素のみの色空間変換及び色分解処理を実行し、その結果をその MCU に対応する全ての画素に反映させることにより、処理の高速化を実現している。

【 0 0 7 8 】

まずステップ S 2 1 で、変数 F l a g を「 0 」に初期化する。尚、この変数 F l a g は、メモリ 1 0 1 4 に記憶されている。次にステップ S 2 2 に進み、この MCU の 1 つ前に処理した MCU の MCU 分類情報 5 2 0 が「 1 」であったかどうかを判定する。そうであればステップ S 2 3 に進み、現 MCU と、1 つ前に処理した MCU の左上の画素値とを比較し、一致するかどうかを判断する。ここで一致する場合はステップ S 2 4 に進み、前述の変数 F l a g に「 1 」をセットしてステップ S 2 5 に進む。一方、ステップ S 2 2 で、この MCU の 1 つ前に処理した MCU の MCU 分類情報が「 1 」でない場合、或はステップ S 2 3 で、1 つ前に処理した MCU と現 MCU の左上の画素値が一致しない場合はステップ S 2 5 に進む。

10

【 0 0 7 9 】

ステップ S 2 5 では、変数 F l a g の値が「 0 」かどうかを判定する。「 0 」であれば画素値の計算が必要となるためステップ S 2 6 に進み、現 MCU の左上の画素に対してのみ、前述の色空間変換と色分解処理を施して色分解データを生成する。次にステップ S 2 7 に進み、ステップ S 2 6 で求めた色分解データをキャッシュしてステップ S 2 8 に進む。このようにステップ S 2 7 までの処理により、この MCU の色分解データを出力するための色処理キャッシュデータを準備することができる。

20

【 0 0 8 0 】

一方ステップ S 2 5 で、変数 F l a g の値が「 1 」であれば「パターン A」の MCU が連続しており、かつ、現 MCU と 1 つ前の MCU とが同じ色であることを表している。このため、1 つ前の MCU の色分解データを出力する際に使用したキャッシュデータを、この現 MCU にもそのまま適用できるため、ステップ S 2 6 , S 2 7 の計算処理をスキップしてステップ S 2 8 に進む。

【 0 0 8 1 】

ステップ S 2 8 , S 2 9 (及びステップ S 3 1 , S 3 2) はそれぞれ、x 方向、y 方向の画素数分のループ処理を示し、こうして現 MCU の 8×8 画素の全画素に対してステップ S 3 0 の処理を行う。即ち、ステップ S 2 9 ~ S 3 1 で、現在の y 方向の座標値で示される x 方向の 8 画素 ($x = 0 \sim x = 7$) にステップ S 3 0 で色分解データを出力する。そして、これらステップ S 2 9 ~ S 3 1 の処理を、ステップ S 2 8 , S 3 2 により、 $y = 0 \sim y = 7$ からなる 8 画素ライン分実行する。こうして、前の MCU で使用した色分解データ、或はステップ S 2 6 , S 2 7 でキャッシュした色分解データを、現 MCU の各画素値として出力する。

30

【 0 0 8 2 】

以上説明した図 9 に示す処理により、「パターン A」の特性を利用して、通常 8×8 の 6 4 画素の全てに対して行うべき色空間変換、色分解処理の実行回数を削減できる。これにより、演算量、メモリアクセス回数を抑えた高速な色分解処理が実現できる。

40

【 0 0 8 3 】

尚、ここでは説明の便宜上、MCU の左上の画素同士を比較した例で説明しているが、MCU に含まれる画素であればどの画素を使用してもよいことは言うまでもない。

【 0 0 8 4 】

図 1 0 は、図 8 のステップ S 1 4 の「パターン B」の MCU に対応する色分解データの生成処理 2 を説明するフローチャートである。「パターン B」は、y 方向の 8 画素が同じであるため、y 方向の 1 列毎に 1 画素のみ色空間変換、色分解処理を実行し、その結果を y 方向の 1 列 8 画素に適用することで高速化を実現する。

【 0 0 8 5 】

まずステップ S 4 1 , S 4 2 (及びステップ S 4 7 , S 4 8) は、それぞれ x 方向、y

50

方向の画素数ループする処理を示している。即ち、現MCUの8×8画素に対し、y方向の8画素、つまり縦1列に対する処理をx方向に8回繰り返すことを意味している。

【0086】

ステップS43では、 $y = 0$ かどうか、即ち、y方向の最初の画素かどうかを判断している。そうであれば、これから処理しようとしているy方向8画素の色分解データを出力するための色処理キャッシュデータがまだ用意されていないことを意味しているためステップS44に進む。ステップS44では、この画素に対して前述の色空間変換、色分解処理を施し、色分解データを生成する。そしてステップS45に進み、ステップS44で求めた色分解データを色処理キャッシュデータとしてキャッシュする。次にステップS46に進み、対象画素の色分解データとして、ステップS45でキャッシュされた色処理キャッシュデータを出力する。

10

【0087】

次のy方向の画素では、yの値は「0」でなくなるためステップS43からステップS46に進み、ステップS45でキャッシュされた色分解データを、その画素値として出力する。この処理を $y = 0 \sim y = 7$ まで実行し、更に、ステップS41, S48に示すように、 $x = 0 \sim x = 7$ の間繰り返す。

【0088】

以上説明したように図10のフローチャートで示す処理により、「パターンB」の特性を利用して、通常8×8の64画素の全てに対して行うべき色空間変換、色分解処理の実行回数を、各列ごとに1回の計算による合計8画素分の計算処理に削減できる。その結果、演算量及びメモリアクセスの回数を抑えた、高速な色分解処理が実現できる。

20

【0089】

尚、ここでは説明の便宜上、最上位の1行分の画素値を計算する場合で説明しているが、他の位置の画素値を抽出して色変換を施し、その結果をキャッシュし、その画素が含まれるy方向の8画素全てに適用するようにしてもよい。

【0090】

図11は、図8のステップS16の「パターンC」のMCUに対応する色分解データの生成処理3を説明するフローチャートである。「パターンC」は、x方向の8画素が同じ画素であるため、x方向の1行毎に1画素のみの色空間変換、色分解処理を実行し、その結果を、x方向の1行の他の画素に適用することで高速化処理を実現している。この処理は、図10のフローチャートの処理において、xとyとを入れ替えることにより実現できるため、以下、簡単に説明する。

30

【0091】

ステップS51, S52(及びステップS57, S58)はそれぞれx方向、y方向の画素数分のループ処理を示している。即ち、現MCUの8×8画素に対し、x方向の8画素、つまり横1行に対する処理をy方向に8回繰り返すことを表している。

【0092】

ステップS53では、 $x = 0$ かどうか、即ち、その行の先頭画素かどうかを判断している。そうであればステップS54に進み、その画素に対して前述の色空間変換、色分解処理を施し、色分解データを生成する。そしてステップS55で、ステップS54で得られた色分解データを色処理キャッシュデータとしてキャッシュする。そしてステップS56で、対象画素の色分解データとして、ステップS55でキャッシュされた色処理キャッシュデータを出力する。

40

【0093】

この図11に示す処理により、「パターンC」の特性を利用し、通常8×8の64画素の全てに対して行うべき色空間変換、色分解処理の実行回数を、各行ごとに1回だけで済み、合計8画素分の計算にまで削減できる。その結果、MCUの各画素値の演算量、メモリアクセス回数を抑えて、高速な色分解処理が実現できる。

【0094】

尚、ここでは説明の便宜上、左1列分の画素値を求めて、その値を他の画素に適用した

50

処理で説明しているが、他の画素値を抽出して色変換を施し、その結果をその画素が含まれるx方向の他の画素全てに適用するようにしても良い。

【0095】

図12は、図8のステップS17の「パターンD」のMCUに対応する色分解データの生成処理4を説明するフローチャートである。「パターンD」は、前述した他のパターンとは異なり、x方向、y方向ともに画素値の連続性が保証されていない。このため、このMCUの8×8画素の全てに対して色空間変換、色分解処理を実行する必要がある。

【0096】

まずステップS61、S62（及びステップS65、S66）はそれぞれx方向、y方向の画素数分のループ処理を示す。このMCUの8×8画素からなる全ての画素数分、ステップS63～S64の処理を繰り返すことを意味している。ステップS63では、該当画素に対して前述の色空間変換、色分解処理を施して色分解データを生成する。そしてステップS64で、その対象画素の色分解データとして、ステップS63で求めた色分解データを出力する。

【0097】

以上説明した処理により、MCUのDCT周波数特性を活用した高速な色分解処理が実現できる。更に図9、図10、図11の処理には共通して次の効果も存在する。

【0098】

前述の説明からも分かる通り、図9、図10、図11の処理は、色処理キャッシュデータを常に1つ保持する構成で実現できる。このため、キャッシュ領域が最小限に抑えられ、複数の色処理キャッシュデータを生成し、キャッシュデータテーブルを保持するような技術よりも低コストで実現できる。更に色処理キャッシュデータを適用すべきかどうかを判断するための処理が不要である。これはMCU分類情報520により、そのMCUのデータパターン（パターンA、B、C、D）が特定付けられ、各パターンごとにどのように色処理キャッシュデータを生成し、それをどの画素に対して適用すべきかを決定できるためである。その結果、従来技術のような、色処理キャッシュデータがヒットするか否かの判断処理が不要となり、更に処理を高速化できる。

【0099】

また更に、MCUを基本単位とする処理を行うため、単純なラスタ単位の処理系に対し、MCU間で発生するキャッシュのミスヒット回数が削減できる。これにより、色処理キャッシュデータの生成回数も最小限に抑えられるという効果がある。

【0100】

次に、前述の高速な色分解データの生成処理を印刷データの生成処理に適用した構成について述べる。

【0101】

先に図4を参照して説明した通り、印刷データの生成は次のようなステップでなされる。

- (1) 印刷が指定された画像ファイルを読み込み、デコードする。
- (2) デコード後の色空間をデバイス固有の色空間へと変換する。
- (3) レイアウト情報に従い変倍する。
- (4) デバイスの各画素の状態を表す量子化量への変換（ハーフトニング）を行う。

【0102】

これらのステップの内、(1)、(2)については前述の構成により高速な処理が実現される。ここでは前述の実施の形態1をもとに、残る(3)、(4)のステップをどのように実現するかについて説明する。

【0103】

以下、図13を参照して、前述の色分解データの生成処理を印刷データの生成処理に適用した構成について説明する。

【0104】

図13は、本発明の他の実施の形態に係る色空間の変換を説明する図である。ここでは

10

20

30

40

50

ハフマン復号化部500乃至色分解データ生成部504までは、前述の図5と同じ構成であり、各ブロックでなされる処理も同じである。

【0105】

変倍部507は、ユーザが指定した用紙サイズ、割り付け方法等のレイアウト指定情報に従って、色分解データ生成部504から出力された色分解データを実印刷サイズに変倍する。ここでの画像データの変倍方法には、ニアレストネイバーや、バイリニア、バイキュービック等様々な手法が存在するが、どの手法を用いてもよい。但し、バイリニアやバイキュービック等の手法を採用する場合は、複数の画素値を用いた補間演算を行うため、MCU単位で処理するのは難しく、変倍部507には少なくとも1MCUラインからなるバンドデータを入力する必要がある。また、ニアレストネイバー等の複数画素の画素値を用いた補間演算を必要としない変倍手法を採用する際は、特に入力データの単位に関する制約はなく、1MCU単位でも、1MCUライン等のバンド単位でも受け渡して処理することが可能である。

10

【0106】

ハーフトーニング部508では、変倍部507で生成された実印刷サイズに変倍された色分解データに対して誤差拡散法等のハーフトーニング手法を施して量子化し出力する。ここで例えばハーフトーニング手法として誤差拡散法を採用する場合、誤差拡散法の誤差を拡散するという処理の性質から、ハーフトーニング部508には少なくとも1ラスタ分のデータが揃っている必要がある。この事情により、予め少なくとも1MCUラインから生成される変倍後のバンドデータを生成しておき、ハーフトーニング部508にラスタデータを入力するようにしなければならない。

20

【0107】

以上の構成により、MCUの周波数特性情報を活用した、高速な色分解データの生成処理による印刷データの生成処理を実現できる。

【0108】

[実施の形態2]

前述の実施の形態1では、MCUの周波数特性を利用してJPEGデコードデータからの色分解データ生成処理を高速化する実施形態について説明した。しかし実施の形態1において、変倍部507の変倍手法としてニアレストネイバー等の複数画素の画素値を用いた補間演算を必要としない変倍手法を採用する場合には、色分解処理だけでなく変倍処理までも含めて高速化が可能となる。これはニアレストネイバー等の複数画素の画素値を用いた補間演算を必要としない変倍手法では、変倍処理をMCU単位で実施できるためである。

30

【0109】

そこで本実施の形態2では、複数画素の画素値を用いた補間演算を必要としない変倍手法を想定した場合に、変倍後の色分解データを高速に生成し、更に、変倍された色分解データから印刷データを生成する形態について説明する。尚、以下の説明では、変倍部507での変倍手法としてニアレストネイバーを採用した場合を想定して説明する。

【0110】

図14は、本発明の実施の形態2に係る色空間の変換を説明する図である。

40

【0111】

図において、ハフマン復号化部500、逆量子化部501、IDCT部502、MCU分類情報生成部503、色空間変換部505、色分解部506、ハーフトーニング部508は前述の図5及び図13に示した各部と等価である。この図14の構成は、前述の実施の形態1とは、変倍後色分解データ生成部509と、変倍部510のみが異なるため、これら2つのブロックに絞って説明する。

【0112】

変倍後色分解データ生成部509は、JPEGデコードデータと、MCU分類情報520とを受け取り、ユーザが指定したレイアウト設定情報により決定される実印刷サイズに変倍された色分解データを高速に生成して出力する。

50

【 0 1 1 3 】

次に図 1 5 ~ 図 1 9 のフローチャートを参照して、具体的に変倍後色分解データ生成部 5 0 9 及び変倍部 5 1 0 でどのような処理を行うのか説明する。

【 0 1 1 4 】

図 1 5 は、本発明の実施の形態 2 に係る画像処理部の変倍後色分解データ生成部 5 0 9 で実行される処理を説明するフローチャートである。ここでは、変倍後色分解データ生成部 5 0 9 に、ある M C U のデコードデータと、その M C U の M C U 分類情報 5 2 0 が入力されたときの処理を示している。尚、この実施の形態 2 に係る画像処理装置 1 2 の構成は前述の実施の形態 1 に係る画像処理装置の構成 (図 2) と同じであるため、その説明を省略する。また、この処理を実行するプログラムはメモリ 1 0 1 4 に記憶されており、C P U 1 0 0 1 a の制御の下に実行される。

10

【 0 1 1 5 】

まずステップ S 7 1 で、現 M C U の M C U 分類情報 5 2 0 が「 1 」かどうかを判定する。「 1 」であればその M C U は「パターン A」であることが分かるためステップ S 7 2 に進み、この性質を利用した図 1 6 に示す変倍後色分解データ生成処理 1 を実行する。尚、この変倍後色分解データ生成処理 1 については図 1 6 のフローチャートを参照して後述する。

【 0 1 1 6 】

ステップ S 7 1 で、M C U 分類情報 5 2 0 が「 1 」でなかった場合はステップ S 7 3 に進み、M C U 分類情報 5 2 0 が「 2 」かどうかを判定する。ここで「 2 」であればステップ S 7 4 に進み、その M C U は「パターン B」であるため、この性質を利用した変倍後色分解データ生成処理 2 を実行する。この変倍後色分解データ生成処理 2 は図 1 7 のフローチャートを参照して後述する。

20

【 0 1 1 7 】

またステップ S 7 3 で、M C U 分類情報 5 2 0 が「 2 」でなかった場合はステップ S 7 5 に進み、M C U 分類情報 5 2 0 が「 3 」かどうかを判定する。ここで「 3 」であれば、この M C U は「パターン C」であるためステップ S 7 6 に進み、この性質を利用した変倍後色分解データ生成処理 3 を実行する。この変倍後色分解データ生成処理 3 は、図 1 8 のフローチャートを参照して後述する。

【 0 1 1 8 】

ステップ S 7 5 で、M C U 分類情報 5 2 0 が「 3 」でなかった場合は、その M C U は「パターン D」であるためステップ S 7 7 に進み、この性質を考慮した変倍後色分解データ生成処理 4 を実行する。この変倍後色分解データ生成処理 4 については図 1 9 に示すフローチャートを参照して後述する。このように図 1 5 のフローチャートで示すように、変倍後色分解データ生成部 5 0 9 は、入力された M C U 分類情報 5 2 0 に従って変倍後色分解データ生成処理方法を切り替えている。

30

【 0 1 1 9 】

次に図 1 6 のフローチャートを参照して、本実施の形態 2 に係る変倍後色分解データ生成処理 1 について説明する。

【 0 1 2 0 】

図 1 6 は、本実施の形態 2 に係る「パターン A」の M C U に対応する色分解データの生成処理 1 (ステップ S 7 2) を説明するフローチャートである。「パターン A」は、M C U 内の 8 × 8 画素の全てが同じ画素値である。よって、この処理では、M C U 内の 1 画素のみの色空間変換、色分解処理を実行し、その結果を全ての画素に反映させることにより、処理の高速化を実現している。

40

【 0 1 2 1 】

図 1 6 は、8 × 8 画素の M C U を W × H 画素に変倍する場合を想定した処理を示している。尚、この変倍後のサイズ W × H は、ニアレストネイバー等の変倍手法に従い算出すればよく、本実施の形態では特筆すべき内容ではないため、ここでは説明を割愛する。

【 0 1 2 2 】

50

まずステップS81で、変数Flagを「0」に初期化する。この変数Flagは、メモリ1014に記憶される。次にステップS82に進み、現MCUの1つ前に処理したMCUのMCU分類情報520の値も「1」であったかどうかを判定する。そうであればステップS83に進み、現MCUと、1つ前に処理したMCUの左上の画素値とを比較し、一致するかどうかを判定する。一致すればステップS84に進み、キャッシュされている色分解データを使用できることを示すように、変数Flagに「1」を代入してステップS85に進む。尚、ステップS82で、1つ前に処理したMCUのMCU分類情報520の値が「1」でない場合、或はステップS83で、1つ前に処理したMCUと現MCUの対応する画素値とが異なる場合はステップS85に進む。

【0123】

ステップS85では、変数Flagの値が「0」かどうかを判断する。ここでFlagの値が「1」であれば「パターンA」のMCUが連続しており、かつ現MCUと1つ前のMCUとが同じ色であることを意味している。よってステップS88に進み、1つ前のMCUの色分解データを出力する際に使用したキャッシュデータを、現MCUにもそのまま適用する。逆に、Flagの値が「0」であればステップS87に進み、現MCUの色分解データを生成するための変換処理を実行する。

【0124】

ステップS86, S87は、キャッシュデータを生成するための処理である。ステップS86では、現MCUの左上の画素に対してのみ、前述の色空間変換と色分解処理を施して色分解データを生成する。そしてステップS87で、ステップS86で求めた色分解データをキャッシュする。このようにステップS86, S87の処理で、このMCUの変倍後色分解データを出力するための色処理キャッシュデータを求めることができる。

【0125】

次にステップS88, S89(及びステップS91, S92)は、それぞれy方向、x方向の画素数分のループ処理を示し、現MCUの変倍後の領域サイズH×Wの画素全てに対してステップS90の処理を行うことを表している。そしてステップS90では、予めキャッシュされていた、或はステップS87でキャッシュした色処理キャッシュデータを該当画素の色分解データとして出力する。

【0126】

このように図16に示す処理により、「パターンA」の特性を利用して、通常8×8の64画素の全てに対して行うべき色空間変換、色分解処理の実行回数を削減できる。これにより、演算量、メモリアクセス回数を抑えた高速な色分解処理が実現できる。

【0127】

更に、前述の実施の形態1に対して、更にメモリアクセス回数を抑えた処理が実現可能となる。これは8×8の64画素分、色分解データを一度メモリ1014に記憶した後に変倍を行うのではなく、ステップS87でキャッシュされたデータを、直接、変倍後の色分解データとして出力するためである。

【0128】

またここでは説明の便宜上、左上の画素に対して色分解データを求める処理で説明しているが、このMCUに含まれる画素であれば、どの位置の画素を使用してもよい。

【0129】

次に図17を参照して、変倍後色分解データ生成処理2について詳細を説明する。

【0130】

図17は、本実施の形態2に係る「パターンB」のMCUに対応する色分解データの生成処理2(ステップS74)を説明するフローチャートである。「パターンB」は、入力データのy方向8画素が同じ画素であるため、y方向1列毎に1画素のみ色空間変換、色分解処理を実行する。そして、その結果を用いて変倍率を考慮したx方向のx行、y方向y列分の画素に適用することで高速化を実現する。例えば、現MCUをニアレストネイバーで16×16画素に変倍する場合、1つの色処理キャッシュデータは2×16の画素領域に適用される。

10

20

30

40

50

【 0 1 3 1 】

図 1 7 は、 8×8 画素の M C U を $W \times H$ 画素に変倍する場合を想定した処理を示すフローチャートである。尚、変倍後のサイズ $W \times H$ は、ニアレストネイバーの変倍手法に従って算出すればよく、本実施の形態 2 では特筆すべき内容ではないため、ここでは説明を割愛する。

【 0 1 3 2 】

まずステップ S 1 0 1 で、変数 O_x を「0」に初期化する。この変数 O_x はメモリ 1 0 1 4 に記憶される。ステップ S 1 0 2 (及びステップ S 1 1 2) は、入力データの x 方向の画素数分のループ処理を示す。ステップ S 1 0 3 , S 1 0 4 により、入力データの y 方向の 1 列毎に 1 画素のみの色空間変換、色分解処理を実行し、その変換結果をキャッシュ

10

【 0 1 3 3 】

ステップ S 1 0 5 は、現在着目している入力画素データの x 方向の繰り返し出力回数 W_x を算出している。この W_x についても前述の W , H と同様に、ニアレストネイバーの変倍手法に従い算出すればよく、本実施の形態 2 では特筆すべき内容ではないため、ここでは説明を割愛する。因みに、 $x = 0 \sim 7$ の W_x の合計は W となる。

【 0 1 3 4 】

具体的な設定値の具体例としては、この M C U をニアレストネイバーで 16×16 画素に変倍する場合、1 つの色処理キャッシュデータは 2×16 の画素領域に適用されるが、この場合全ての x で $W_x = 2$ となる。

20

【 0 1 3 5 】

次にステップ S 1 0 6 , S 1 0 7 (及びステップ S 1 0 9 , S 1 1 0) は、それぞれ変倍出力データの x 方向、 y 方向の画素数分のループ処理を示す。ステップ S 1 0 8 は、ステップ S 1 0 4 までの処理で得られた色処理キャッシュデータを変倍後色分解データとして、 $(O_x, 0)$ を左上位置座標とする $W_x \times H$ サイズの矩形領域に出力することを意味している。またステップ S 1 1 1 では、変倍後の色分解データの出力位置を更新するために O_x に W_x を加えている。

【 0 1 3 6 】

以上図 1 7 に示す処理により、「パターン B」の特性を利用して、通常 8×8 の 64 画素の全てに対して行うべき色空間変換、色分解処理の実行回数を、各列 1 画素だけの合計 8 画素分にまで削減できる。この結果、演算量、メモリアクセス回数を抑えた高速な色分解処理が実現できる。

30

【 0 1 3 7 】

更に、前述の実施の形態 1 よりもメモリアクセス回数を抑えた処理が実現可能となる。これは 8×8 の 64 画素分、色分解データを一度メモリに出力した後に変倍を行うのではなく、ステップ S 1 0 4 でキャッシュされたデータを、直接変倍後のデータとして出力するためである。尚、ここでは説明の便宜上、上 1 行分の画素に対して計算する処理で説明しているが、他の位置の画素を抽出して色変換を施し、その結果をキャッシュしても良い。

【 0 1 3 8 】

図 1 8 は、本実施の形態 2 に係る「パターン C」の M C U に対応する色分解データの生成処理 3 (ステップ S 7 6) を説明するフローチャートである。「パターン C」は、入力データの x 方向 8 画素が同じ画素値であるため、 x 方向の 1 行毎に 1 画素のみ色空間変換、色分解処理を実行する。そして、その結果を、変倍率を考慮した x 方向の x 行、 y 方向 y 列分の画素に適用することで高速化を実現する。例えば現 M C U をニアレストネイバーで 16×16 画素に変倍する場合、1 つの色処理キャッシュデータは 16×2 の画素領域に適用される。

40

【 0 1 3 9 】

図 1 8 は、 8×8 画素の M C U を $W \times H$ 画素に変倍する場合を想定したフローチャートを示している。尚、変倍後のサイズ $W \times H$ は、ニアレストネイバーの変倍手法に従って算

50

出すればよく本実施の形態において特筆すべき内容ではないため、ここでは説明を割愛する。

【0140】

まずステップS121で、変数Oyを「0」に初期化する。この変数Oyはメモリ1014に記憶される。次にステップS122（及びS132）は、入力データのy方向の画素数分のループ処理を示す。ステップS123，S124により、入力データのx方向の1行毎に1画素のみの色空間変換、色分解処理を実行し、その変換結果をキャッシュすることを意味している。

【0141】

ステップS125は、現在着目している入力画素データのy方向の繰り返し出力回数Hyを算出している。このHyについても前述のW，Hと同様に、ニアレストネイバーの変倍手法に従い算出すればよい。因みに、 $y = 0 \sim 7$ のHyの合計はHとなる。

10

【0142】

具体的な設定値の例では、MCUをニアレストネイバーで 16×16 画素に変倍する場合、1つの色処理キャッシュデータは 16×2 の画素領域に適用されるが、この場合は全てのyでHy = 2となる。

【0143】

次にステップS126，S127（及びS129，S130）は、それぞれ変倍出力データのy方向、x方向の画素数分のループ処理を示す。ステップS128は、ステップS124までの処理で得られた色処理キャッシュデータを変倍後の色分解データとして、(0, Oy)を左上位置座標とする $W \times Hy$ サイズの矩形領域に出力することを意味している。ステップS131は、変倍後の色分解データの出力位置を更新するためにOyにHyを加えている。

20

【0144】

以上説明した図18に示す処理により、「パターンC」の特性を利用して、通常 8×8 の64画素の全てに対して行うべき色空間変換、色分解処理の実行回数を、各行ごとに1画素の合計8画素分（8回）にまで削減できる。その結果、演算量、メモリアクセス回数を抑えた高速な色分解処理が実現できる。

【0145】

更に、前述の実施の形態1よりもメモリアクセス回数を抑えた処理が実現可能となる。これは 8×8 の64画素分、色分解データを一度メモリ1014に出力した後に変倍を行うのではなく、ステップS124でキャッシュされたデータを、直接、変倍後のデータとして出力するためである。

30

【0146】

尚、ここでは説明の便宜上、左1列分の画素を計算する場合で説明しているが、他の位置の画素を抽出して色変換を施し、その結果をキャッシュしても良い。

【0147】

次に図19を参照して、変倍後色分解データ生成処理4（ステップS77）について詳細を説明する。

【0148】

図19は、本実施の形態2に係る「パターンD」のMCUに対応する色分解データの生成処理4を説明するフローチャートである。「パターンD」は、他のパターンとは異なり、x方向、y方向に画素値の連続性が保証されていない。このため、該当MCUの 8×8 画素の全てに対して色空間変換、色分解処理を実行し、その変換結果をキャッシュして、各画素の変倍後サイズを考慮して出力する。

40

【0149】

図19は、 8×8 画素のMCUを $W \times H$ 画素に変倍する場合を想定した処理を示している。尚、変倍後のサイズ $W \times H$ はニアレストネイバーの変倍手法に従い算出すればよく、本実施の形態2においては特筆すべき内容ではないため、ここでは説明を割愛する。

【0150】

50

まずステップS 1 4 1で、変数O xを「0」に初期化する。この変数O xは図17のフローチャートのステップS 1 0 1の変数と同様である。ステップS 1 4 2（及びS 1 5 7）は、入力データのx方向の画素数分のループ処理を示す。

【0151】

ステップS 1 4 3は、現在着目している入力画素データのx方向の繰り返し出力回数W xを算出している。このW xについても前述のW、Hと同様に、ニアレストネイバーの変倍手法に従い算出すればよい。因みに、 $x = 0 \sim 7$ のW xの合計はWとなる。

【0152】

次にステップS 1 4 4に進み、変数O yを「0」に初期化する。この変数O yは図18のフローチャートのステップS 1 2 1の変数と同様である。ステップS 1 4 5（及びS 1 5 5）は、入力データのy方向の画素数分のループ処理を示す。

10

【0153】

次にステップS 1 4 6で、入力データである位置(x, y)の画素に対する色空間変換、色分解処理を実行する。そしてステップS 1 4 7で、その変換結果をキャッシュする。次にステップS 1 4 8は、現在着目している入力画素データのy方向の繰り返し出力回数H yを算出している。このH yについても前述のW、Hと同様に、ニアレストネイバーの変倍手法に従い算出すればよい。因みに、 $y = 0 \sim 7$ のH yの合計はHとなる。

【0154】

次にステップS 1 4 9, S 1 5 0（及びS 1 5 2, S 1 5 3）は、それぞれ変倍出力データのx方向、y方向の画素数分のループ処理を示す。次にステップS 1 5 1では、ステップS 1 4 7までの処理で得られた色処理キャッシュデータを変倍した後、色分解データとして、(O x, O y)を左上位置座標とするW x x H yサイズの矩形領域に出力する。そしてステップS 1 5 4では、変倍後の色分解データの出力位置を更新するために、変数O yにH yを加えている。同様にステップS 1 5 6でも、変倍後の色分解データの出力位置を更新するために変数O xにW xを加えている。

20

【0155】

以上説明したように図19に示す処理により、「パターンD」の特性に従う処理が実現される。上述の通り、「パターンD」の特性のため、色空間変換、色分解処理の実行回数については削減できない。しかし、前述の実施の形態1よりもメモリアクセス回数を抑えた処理が実現可能となる。これは8 x 8の64画素分、色分解データを一度メモリ1014に出力した後に変倍を行うのではなく、ステップS 1 4 7でキャッシュされたデータを直接変倍後データとして出力するためである。

30

【0156】

以上、説明した処理により、MCUのDCT周波数特性を活用した高速な変倍後色分解データの生成処理が実現できる。更に図16~図19の処理には共通して次の効果も存在する。

【0157】

前述の説明からも分かる通り、図16~図19の処理は、色処理キャッシュデータを常に1つ保持する構成で実現できる。このため、キャッシュ領域を最小限に抑えることができる。これにより複数の色処理キャッシュデータを生成し、キャッシュデータテーブルを保持するような技術よりも低コストで実現可能である。

40

【0158】

更に、色処理キャッシュデータを適用すべきかどうかを判断するための処理が不要となる。これはMCU分類情報520により、そのMCUのデータパターン（パターンA, B, C, D）を特定付けられることによる。即ち、データパターンごとに、どのように色処理キャッシュデータを生成し、それをどの画素に対して適用すべきかが決定できる。その結果、従来技術のような色処理キャッシュデータがヒットするか否かの判断処理が不要となり、更なる高速化につながる。

【0159】

更に、MCUを基本単位とする処理を行うため、単純なラスタ単位の処理系に対し、M

50

ＣＵ間で発生するキャッシュのミスヒット回数が削減でき、それに伴う色処理キャッシュデータの生成回数も最小限に抑えられる。

【 0 1 6 0 】

以上説明した上記実施の形態では、ホスト装置と画像処理装置が接続されたシステム、もしくは画像処理装置にメモリカードが直接接続されたシステムについて記述した。しかし本発明は、複数機器の組み合わせから構成されるシステムであっても、ＤＣＴに基づき圧縮された画像データが入力され、その画像データから印刷データを生成するようなシステムにも適用可能であることは言うまでもない。

【 0 1 6 1 】

また、本発明の目的は前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体（または記録媒体）を、システムあるいは装置に供給しても達成されることは言うまでもない。この場合、そのシステムあるいは装置のコンピュータ（またはＣＰＵやＭＰＵ）が記憶媒体に格納されたプログラムコードを読み出し実行することによって目的が達成される。この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでない。そのプログラムコードの指示に基づき、コンピュータ上で稼働しているオペレーティングシステム（ＯＳ）等が実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。

【 0 1 6 2 】

更に、以下のような場合も前述した実施形態の機能が実現される。記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張カードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれる。その後、そのプログラムコードの指示に基づき、その機能拡張カードや機能拡張ユニットに備わるＣＰＵ等が実際の処理の一部または全部を行うような場合である。

【 0 1 6 3 】

前述の実施形態では印刷データ生成処理に絞り説明した。しかし本発明はＪＰＥＧのようなＤＣＴに基づき圧縮された画像データに対し、画素単位で他の色空間への変換等、何かしらの変換処理を施すような技術に対しても適用可能である。例えばＪＰＥＧ画像をディスプレイに表示させる際のカラーマッチング処理等にも適用可能である。

【 0 1 6 4 】

以上説明したように本実施の形態によれば、ＤＣＴに基づき圧縮されたカラー画像データに対する色処理について、演算量の削減、メモリアクセス回数の削減が実現でき、高速化が実現できる。

【 0 1 6 5 】

また本実施の形態では、上述の特許文献１と同様に、ＤＣＴ係数の特性に着目して処理を軽減しているが、特許文献１の技術に対し、単一色のＭＣＵにとどまらず、非単一色のＭＣＵに対しても処理の効率化を実現している。

【 0 1 6 6 】

また本実施の形態では、色処理キャッシュを行うが、特許文献２で提案されている技術に対して、より低コストで実現できる。これはキャッシュデータテーブルを生成する必要がなく、常に１組の演算結果だけをキャッシュしているためである。これによりキャッシュ領域のサイズを小さくでき、コスト面でも秀でている。

【 0 1 6 7 】

更に本実施の形態では、ＤＣＴ周波数特性に基づいてＭＣＵを分類してデータをパターン化しているため、キャッシュデータテーブルの検索が不要であり、無駄を排除した良好な処理が実現できる。

【 0 1 6 8 】

更に、ＭＣＵを意識した処理を行うため、単純なラスタ処理を行う技術に対して、ＭＣ

10

20

30

40

50

U間で発生するキャッシュのミスヒット回数を削減でき、キャッシュのヒット率を向上することができる。

【図面の簡単な説明】

【0169】

【図1】本発明の実施の形態に共通するプリントシステム（画像処理システム）の概略構成を示すブロック図である。

【図2】本実施の形態に係るプリントシステムを構成する画像処理装置のハードウェアの概略を示すブロック図である。

【図3】本実施の形態に係る画像処理装置のプリント機能で用いられるソフトウェアのモジュール構造を示す図である。

【図4】本実施の形態に係るデータ変換モジュールによる印刷データの生成処理を説明するフローチャートである。

【図5】本実施の形態に係る色空間の変換処理を説明する図である。

【図6】本実施の形態に係る逆量子化部により得られるDC T係数を説明する図である。

【図7】IDCTにより得られる画素値を説明する図である。

【図8】本発明の実施の形態1に係る色分解データ生成部で実行される処理を説明するフローチャートである。

【図9】実施の形態1に係る「パターンA」のMCUに対応する色分解データの生成処理1を説明するフローチャートである。

【図10】実施の形態1に係る「パターンB」のMCUに対応する色分解データの生成処理2を説明するフローチャートである。

【図11】実施の形態1に係る「パターンC」のMCUに対応する色分解データの生成処理3を説明するフローチャートである。

【図12】実施の形態1に係る「パターンD」のMCUに対応する色分解データの生成処理4を説明するフローチャートである。

【図13】本発明の実施の1形態の他の実施例に係る色空間の変換処理を説明する図である。

【図14】本発明の実施の形態2に係る色空間の変換処理を説明する図である。

【図15】本発明の実施の形態2に係る画像処理部の色分解データ生成部で実行される処理を説明するフローチャートである。

【図16】本実施の形態2に係る「パターンA」のMCUに対応する色分解データの生成処理1を説明するフローチャートである。

【図17】本実施の形態2に係る「パターンB」のMCUに対応する色分解データの生成処理2を説明するフローチャートである。

【図18】本実施の形態2に係る「パターンC」のMCUに対応する色分解データの生成処理3を説明するフローチャートである。

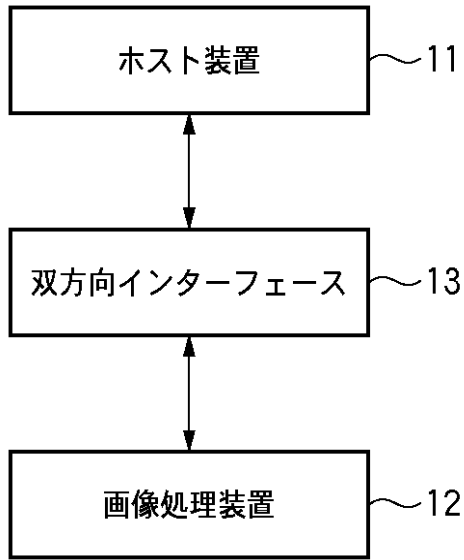
【図19】本実施の形態2に係る「パターンD」のMCUに対応する色分解データの生成処理3を説明するフローチャートである。

10

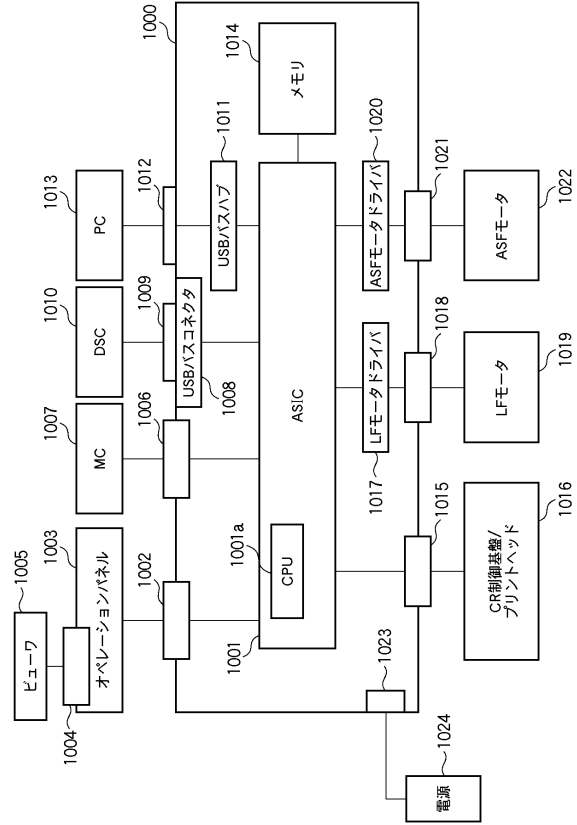
20

30

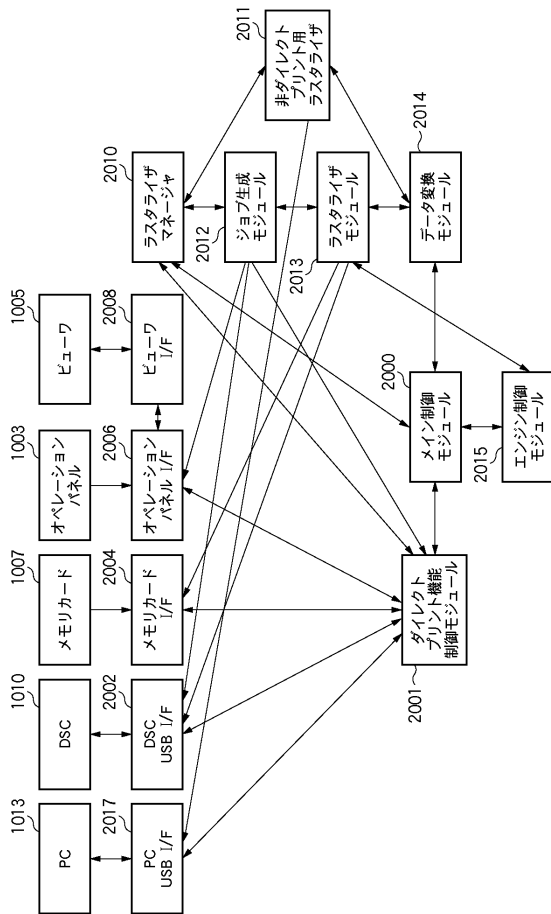
【図1】



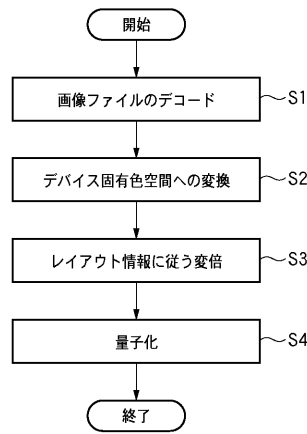
【図2】



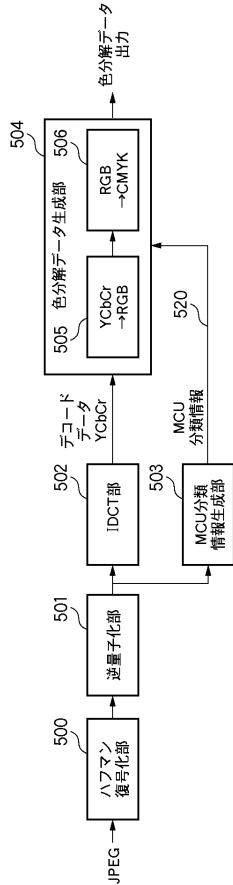
【図3】



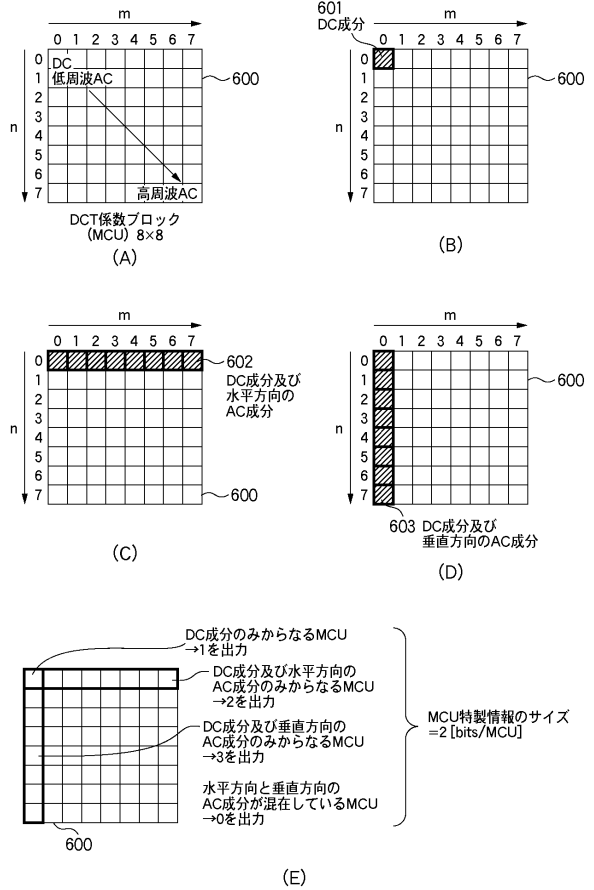
【図4】



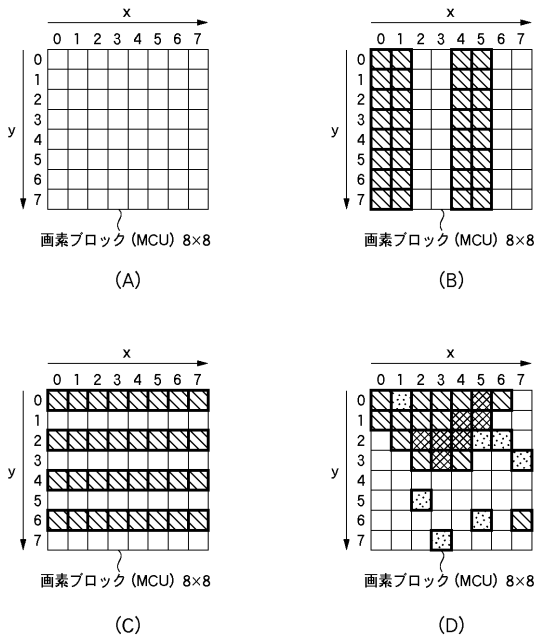
【図5】



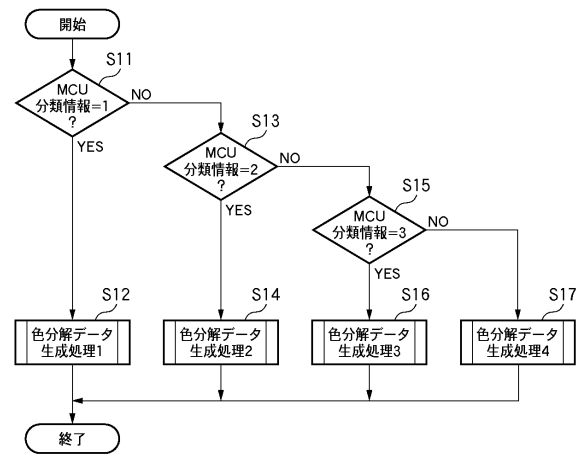
【図6】



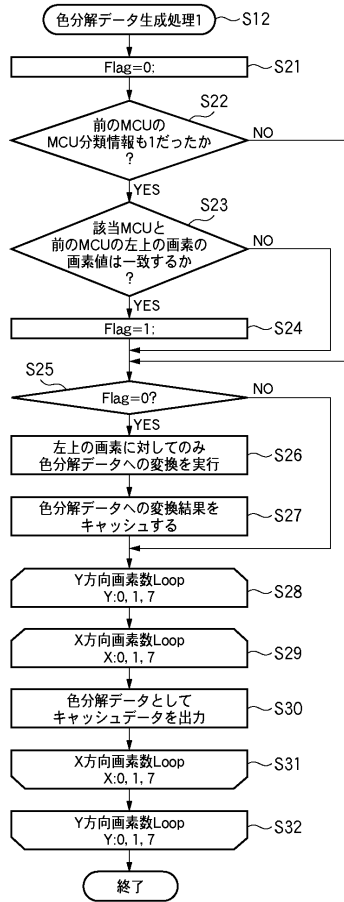
【図7】



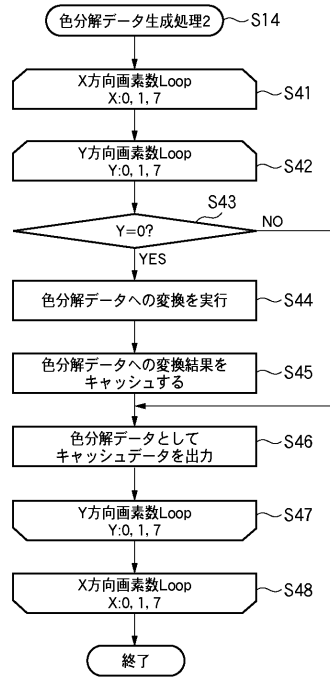
【図8】



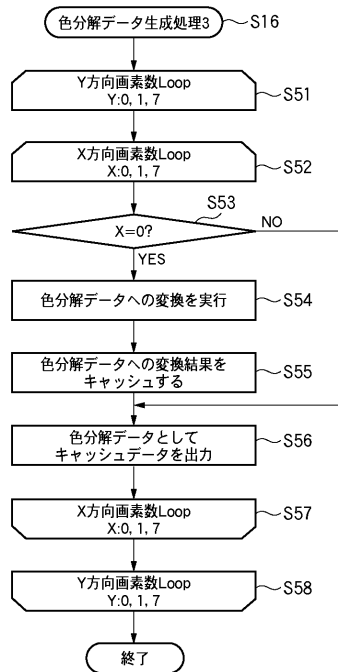
【図9】



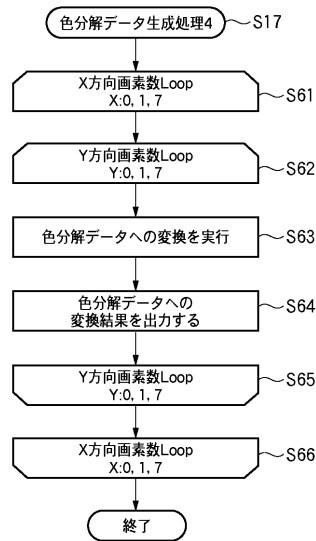
【図10】



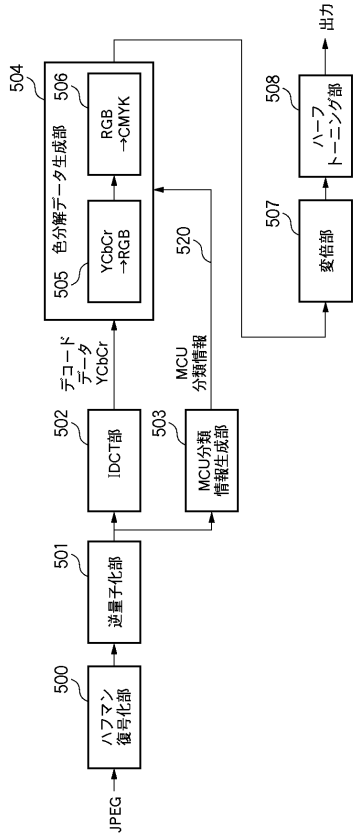
【図11】



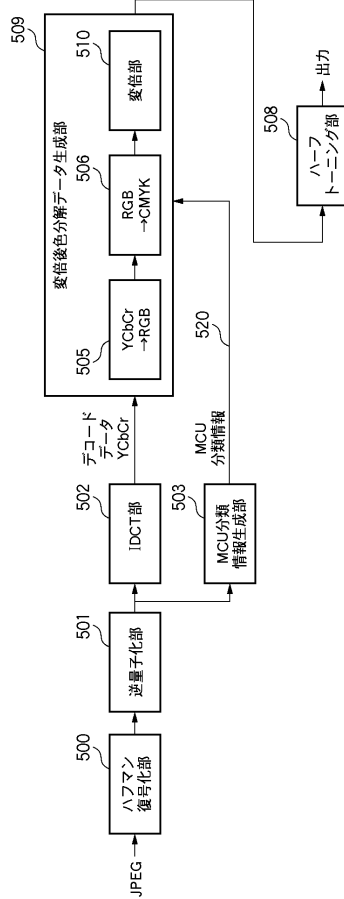
【図12】



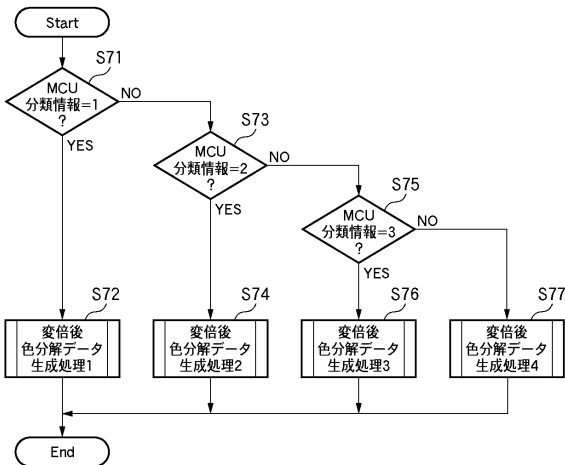
【図13】



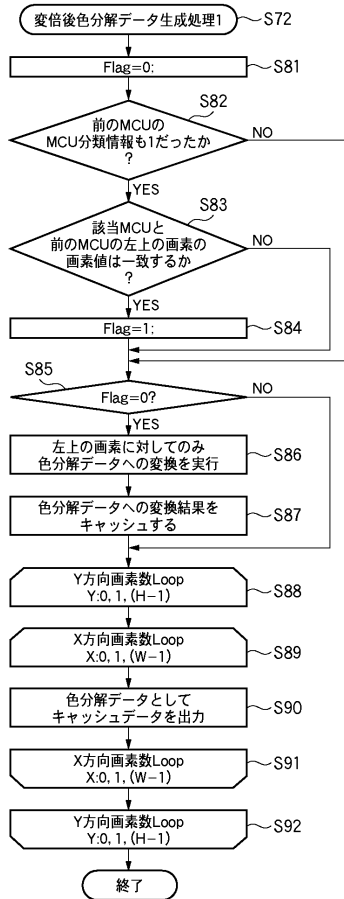
【図14】



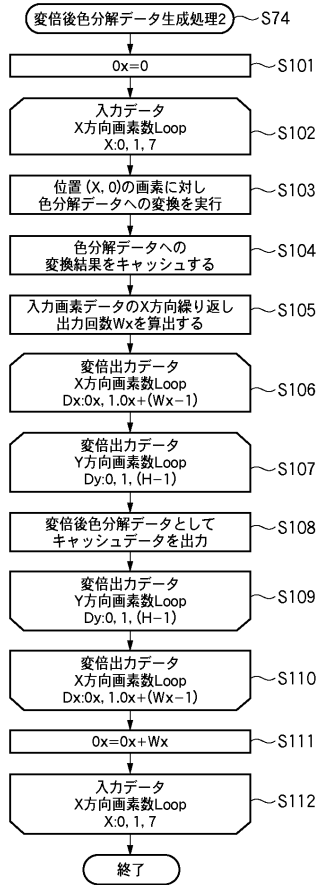
【図15】



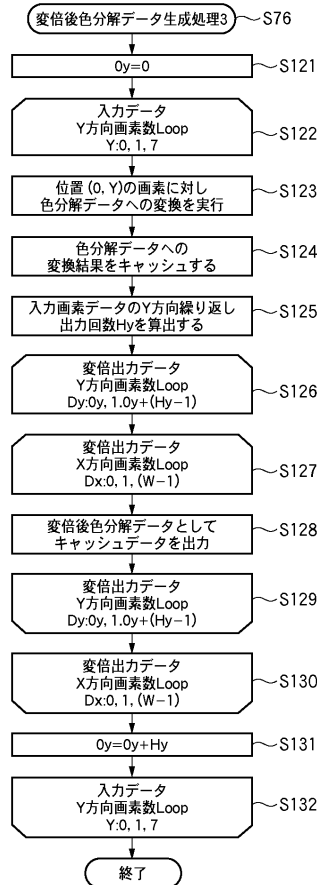
【図16】



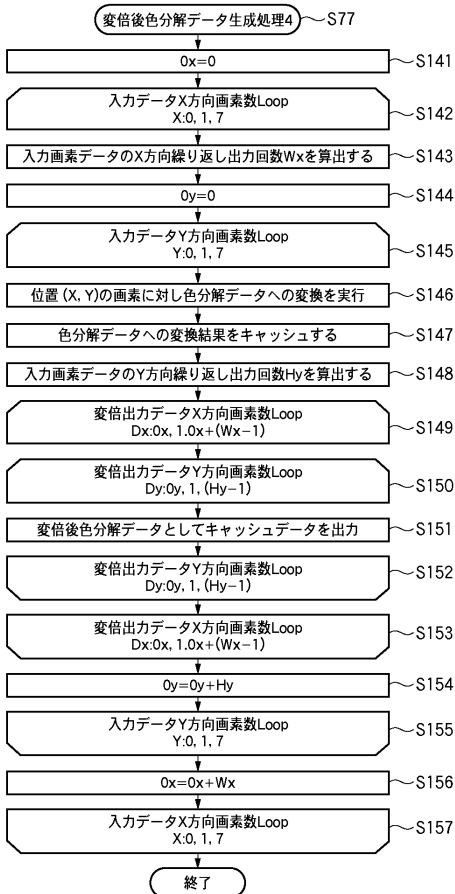
【図17】



【図18】



【図19】



フロントページの続き

審査官 富永 達朗

(56)参考文献 特開平8 - 265755 (JP, A)
特開2001 - 145107 (JP, A)

(58)調査した分野(Int.Cl., DB名)
H04N 1/41
H04N 1/46
H04N 1/60