

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5383416号
(P5383416)

(45) 発行日 平成26年1月8日(2014.1.8)

(24) 登録日 平成25年10月11日(2013.10.11)

(51) Int. Cl. F I
G06T 5/20 (2006.01) G O 6 T 5/20 A
H04N 1/40 (2006.01) H O 4 N 1/40 I O I Z

請求項の数 10 (全 25 頁)

(21) 出願番号	特願2009-232862 (P2009-232862)	(73) 特許権者	000001007
(22) 出願日	平成21年10月6日 (2009.10.6)		キヤノン株式会社
(65) 公開番号	特開2011-81573 (P2011-81573A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成23年4月21日 (2011.4.21)	(74) 代理人	100076428
審査請求日	平成24年10月4日 (2012.10.4)		弁理士 大塚 康德
		(74) 代理人	100112508
			弁理士 高柳 司郎
		(74) 代理人	100115071
			弁理士 大塚 康弘
		(74) 代理人	100116894
			弁理士 木村 秀二
		(74) 代理人	100130409
			弁理士 下山 治
		(74) 代理人	100134175
			弁理士 永川 行光

最終頁に続く

(54) 【発明の名称】 画像処理装置及びその制御方法とプログラム

(57) 【特許請求の範囲】

【請求項1】

画像を入力する入力手段と、前記入力手段により入力された画像を分割して、それぞれがオーバーラップしない領域とオーバーラップ領域とを含む複数のタイル画像を生成する生成手段と、前記生成手段により生成された複数のタイル画像それぞれに対して別々に画像処理を実行する画像処理手段とを有する画像処理装置であって、

前記生成手段は、前記タイル画像を生成する際に、前記オーバーラップ領域の画像の解像度が前記オーバーラップしない領域の画像の解像度より小さくなるように、前記オーバーラップ領域の画像の解像度を変換することを特徴とする画像処理装置。

【請求項2】

前記生成手段は、前記オーバーラップしない領域の画像の解像度を変換せずに、前記オーバーラップ領域の画像の解像度を小さくするように変換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】

前記画像に基づいて印刷を実行する際の印刷モードを設定する設定手段を更に有し、前記生成手段は、前記オーバーラップ領域の画像の解像度を前記設定手段により設定された印刷モードに応じた解像度に変換することを特徴とする請求項1又は2に記載の画像処理装置。

【請求項4】

前記オーバーラップ領域は複数のサブ領域を含み、前記生成手段は、前記複数のサブ領

域の画像の解像度を領域ごとに異なる解像度に変換することを特徴とする請求項 1 乃至 3 の何れか 1 項に記載の画像処理装置。

【請求項 5】

画像を入力する入力手段と、前記入力手段により入力された画像を分割して、それぞれがオーバーラップしない領域とオーバーラップ領域とを含む複数のタイル画像を生成する生成手段と、前記生成手段により生成された複数のタイル画像それぞれに対して別々に画像処理を実行する画像処理手段とを有する画像処理装置であって、

前記生成手段は、前記タイル画像を生成する際に、前記オーバーラップ領域の画像が前記オーバーラップしない領域の画像より高い圧縮率で圧縮されるように、前記オーバーラップ領域の画像を圧縮することを特徴とする画像処理装置。

10

【請求項 6】

前記生成手段は、前記オーバーラップしない領域の画像を第 1 の圧縮率で圧縮して、前記オーバーラップ領域の画像を前記第 1 の圧縮率より高い第 2 の圧縮率で圧縮することを特徴とする請求項 5 に記載の画像処理装置。

【請求項 7】

画像処理装置の制御方法であって、

画像を入力する入力工程と、

前記入力工程により入力された画像を分割して、それぞれがオーバーラップしない領域とオーバーラップ領域とを含む複数のタイル画像を生成する生成工程と、

前記生成工程により生成された複数のタイル画像それぞれに対して別々に画像処理を実行する画像処理工程とを有し、

20

前記生成工程では、前記タイル画像を生成する際に、前記オーバーラップ領域の画像の解像度が前記オーバーラップしない領域の画像の解像度より小さくなるように、前記オーバーラップ領域の画像の解像度を変換することを特徴とする制御方法。

【請求項 8】

画像処理装置の制御方法であって、

画像を入力する入力工程と、

前記入力工程により入力された画像を分割して、それぞれがオーバーラップしない領域とオーバーラップ領域とを含む複数のタイル画像を生成する生成工程と、

前記生成工程により生成された複数のタイル画像それぞれに対して別々に画像処理を実行する画像処理工程とを有し、

30

前記生成工程では、前記タイル画像を生成する際に、前記オーバーラップ領域の画像が前記オーバーラップしない領域の画像より高い圧縮率で圧縮されるように、前記オーバーラップ領域の画像を圧縮することを特徴とする制御方法。

【請求項 9】

画像処理装置に、

画像を入力する入力工程と、

前記入力工程により入力された画像を分割して、それぞれがオーバーラップしない領域とオーバーラップ領域とを含む複数のタイル画像を生成する生成工程と、

前記生成工程により生成された複数のタイル画像それぞれに対して別々に画像処理を実行する画像処理工程とを実行させるためのプログラムであって、

40

前記生成工程では、前記タイル画像を生成する際に、前記オーバーラップ領域の画像の解像度が前記オーバーラップしない領域の画像の解像度より小さくなるように、前記オーバーラップ領域の画像の解像度を変換することを特徴とするプログラム。

【請求項 10】

画像処理装置に、

画像を入力する入力工程と、

前記入力工程により入力された画像を分割して、それぞれがオーバーラップしない領域とオーバーラップ領域とを含む複数のタイル画像を生成する生成工程と、

前記生成工程により生成された複数のタイル画像それぞれに対して別々に画像処理を実

50

行する画像処理工程とを実行させるためのプログラムであって、

前記生成工程では、前記タイル画像を生成する際に、前記オーバーラップ領域の画像が前記オーバーラップしない領域の画像より高い圧縮率で圧縮されるように、前記オーバーラップ領域の画像を圧縮することを特徴とするプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、画像データに所定の画像処理を施して出力をする画像処理装置及びその制御方法とプログラムに関するものである。

【背景技術】

【0002】

複写機、ファクシミリ、レーザプリンタ等の画像処理装置では、入力したデジタル画像の各ラインをラインメモリに格納し、回転や各種フィルタ処理等を行っている。これにより画像データの回転、平滑化、拡大/縮小、更にはエッジ強調等の画像処理を実行している。このような画像処理を実行する画像処理装置は一般に、ページメモリや画像処理や画像の編集用に、回転部、フィルタ処理部等を含む画像処理部を有している。このような画像処理部で画像データを、例えば90度回転する回転部の場合で説明する。画像データを90度回転する場合は、例えばページメモリに画像データをラスタ（水平方向の走査順）順に1ページ分の画像データを書き込む。そして、その画像データを読み出す際には、垂直方向に読み出すことにより画像データの回転を実現できる。しかしこのような回転を実現するためには、1ページの最大サイズに対応する巨大なページメモリが必要となる。

【0003】

またページメモリの容量の増大を抑制する手法として、ページ画像をタイル画像に分割し、各タイルごとに画像データを回転し、各回転した各タイル画像を最終段で結合することによりページ画像を回転する手法がある。

【0004】

しかしながらこの手法では、注目画素と、その注目画素に対する周辺画素を参照して注目画素のフィルタ演算を行う場合、タイル画像の境界の画素が注目画素となる場合は、1つのタイル画像だけで演算を実行することができない。従って、タイル画像の境界の画素が注目画素となる場合は、周辺画素を含めて演算するためにフィルタ演算のサイズに応じてバンドメモリを持たなければならない。

【0005】

このような周辺画素を参照する演算に対する解決手法として、処理対象の原画像をタイル画像に分割する際に、このタイル画像の境界近傍部分の注目画素が周辺画素を参照できるように画素をオーバーラップさせて分割する方法がある。そして周辺画素を用いてタイル画像の境界の画素を処理する場合には、上述したような周辺画素を保持するためのラインメモリは必要とせず、このオーバーラップ部分の画素を参照してフィルタ演算等を行っている（特許文献1参照）。

【先行技術文献】

【特許文献】

【0006】

【特許文献1】特開2005-198121号公報

【発明の概要】

【発明が解決しようとする課題】

【0007】

しかしながら、上述のようにタイル画像に分割する際にオーバーラップ部を付与して画像を分割すると、各タイル画像のデータ量は、本身（オーバーラップでない部分）の画像データ量と、オーバーラップ部分の画像データ量とを加算したデータ量となる。例えば、タイル画像が16画素×16画素（256画素）の場合、オーバーラップ部である周辺画素を2画素を付与すると、各タイル画像は20画素×20画素（400画素）となる。即

10

20

30

40

50

ち、オーバーラップを付与した場合のタイル画像のデータ量は、オーバーラップを付与しない場合に比べて約 1.5 倍となり、タイル画像をメモリから読み出す場合のメモリに必要とされるスループット（必要帯域）も 1.5 倍となる。従って、要求されるスループットが足りない場合には、メモリ仕様として周波数の増加や、バス幅の増加等を行わなければならない、メモリコストの増大が避けられないという課題があった。

【0008】

本発明の目的は、上記従来技術の問題点を解決することにある。

【0009】

本発明の特徴は、オーバーラップ領域の画像のデータ量を少なくして画像処理を実現できる画像処理装置及びその制御方法とプログラムを提供することにある。

10

【課題を解決するための手段】

【0010】

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、

画像を入力する入力手段と、前記入力手段により入力された画像を分割して、それぞれがオーバーラップしない領域とオーバーラップ領域とを含む複数のタイル画像を生成する生成手段と、前記生成手段により生成された複数のタイル画像それぞれに対して別々に画像処理を実行する画像処理手段とを有する画像処理装置であって、

前記生成手段は、前記タイル画像を生成する際に、前記オーバーラップ領域の画像の解像度が前記オーバーラップしない領域の画像の解像度より小さくなるように、前記オーバーラップ領域の画像の解像度を変換することを特徴とする。

20

【発明の効果】

【0011】

本発明によれば、オーバーラップ領域の画像のデータ量を少なくして画像処理を実現できる。これにより、メモリのスループットの負担を小さくし、画像処理のコストパフォーマンスを高めることができる。

【図面の簡単な説明】

【0012】

【図1】実施形態に係る画像処理装置の全体構成を示すブロック図である。

【図2】画像処理装置を制御するソフトウェアモジュールの構成図である。

30

【図3】実施形態に係るプリンタ画像処理部の構成を示すブロック図である。

【図4】実施形態1のアドレス制御部が生成するアドレスを説明する図である。

【図5】画像処理装置におけるプリント動作を説明するフローチャートである。

【図6】アドレス制御部のアドレス制御を示すフローチャートである。

【図7】5×5フィルタ演算部による注目画素D22の演算処理の説明図である。

【図8】5×5フィルタ演算部の内部構造を示すブロック図である。

【図9】フィルタ演算に使用するフィルタ係数の一例図である。

【図10】注目画素D6のフィルタ演算を説明する図である。

【図11】プリンタドライバにおける画面表示の一例を示す図である。

【図12】実施形態2に係る画像処理装置のプリント動作を説明するフローチャートである。

40

【図13】実施形態2に係るプリンタ画像処理部の構成を示すブロック図である。

【図14】実施形態2に係るアドレス制御部のアドレス制御を説明する図である。

【図15】アドレス制御部がタイル画像の読出しを行うためのアドレス制御を示すフローチャートである。

【図16】実施形態2における注目画素D24のフィルタ演算を説明する図である。

【図17】3×3のフィルタ演算に使用するフィルタ係数の一例を示した図である。

【図18】実施形態2における注目画素D6のフィルタ演算を説明する図である。

【図19】実施形態2のアドレス制御部が生成するアドレスを説明する図である。

【図20】フィルタ演算部の注目画素D19のフィルタ演算を説明する図である。

50

- 【図 2 1】フィルタ演算部の注目画素 D3 のフィルタ演算を説明する図である。
 【図 2 2】実施形態 3 の画像圧縮部及び画像伸張部の構成を示すブロック図である。
 【図 2 3】実施形態 3 の画像圧縮部による圧縮処理を示すフローチャートである。
 【図 2 4】実施形態 3 の M C U を示す図である。
 【図 2 5】実施形態 3 の量子化の例を示す図である。
 【図 2 6】実施形態 3 に係る量子化部の構成を示すブロック図である。
 【図 2 7】量子化テーブルの例を示す図である。
 【図 2 8】実施形態 3 に係る画像伸張部の処理を説明するフローチャートである。
 【図 2 9】画像伸張部の逆量子化部の構成を示すブロック図である。
 【図 3 0】逆量子化後の値の一例を示す図である。

10

【発明を実施するための形態】
 【0013】

以下、添付図面を参照して本発明の実施形態を詳しく説明する。尚、以下の実施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。

【0014】

[実施形態 1]

図 1 は、本実施形態に係る画像処理装置（複合機）100 の全体構成を示すブロック図である。

【0015】

20

図 1 において、画像処理装置 100 は、画像入力デバイスであるスキャナ 101 や、プリンタ画像処理部 119 で処理された画像データに基づいて印刷するプリンタエンジン 102 を有している。また画像処理装置 100 は、LAN 10 や公衆回線 104 と接続して、画像情報やデバイス情報を入出力する。CPU 105 は、この画像処理装置 100 全体を制御するための中央処理装置である。RAM 106 は、CPU 105 が動作するためのワークメモリを提供しており、また入力した画像データを一時記憶するための画像メモリを提供している。ROM 107 はブート ROM であり、システムのブートプログラムが格納されている。HDD 108 はハードディスクドライブであり、各種処理のためのシステムソフトウェア及び入力された画像データ等を格納する。操作部 I/F 109 は、画像データ等を表示可能な表示画面や操作ボタン等を有する操作部 110 のインタフェース部である。ネットワークインターフェース 111 は、例えば LAN カード等で実現され、LAN 10 を介して外部装置との間で情報の入出力を行う。モデム 112 は公衆回線 104 を介して外部装置との間で情報の入出力を行う。以上のユニットがシステムバス 113 上に配置されている。

30

【0016】

イメージバス I/F 114 は、システムバス 113 と画像データを高速で転送する画像バス 115 とを接続するためのインタフェースであり、データ構造を変換するバスブリッジである。画像バス 115 には、ラスティメージプロセッサ (RIP) 部 116、デバイス I/F 117、スキャナ画像処理部 118、編集用画像処理部 120、画像圧縮部 103、画像伸張部 121、カラーマネージメントモジュール (CMM) 130 が接続されている。RIP 部 116 は、ページ記述言語 (PDL) コードや後述するベクトルデータをイメージに展開する。デバイス I/F 部 117 は、画像伸張部 121 とプリンタ画像処理部 119 を介してスキャナ 101 やプリンタエンジン 102 と画像処理装置 100 とを接続し、画像データの同期系 / 非同期系の変換を行う。また、スキャナ画像処理部 118 は、スキャナ 101 から入力した画像データに対して、補正、加工、編集等の各種処理を行う。編集用画像処理部 120 は、画像データの回転や、トリミング・マスキング等の各種画像処理を行う。画像圧縮部 103 はスキャナ画像処理部 118 や編集用画像処理部 120 で処理された画像データを HDD 108 で格納する際に所定の圧縮方式で符号化する。画像伸張部 121 は、HDD 108 に格納されている圧縮画像データを復号して伸張する。プリンタ画像処理部 119 は、プリンタエンジン 102 に出力する画像データに、その

40

50

プリンタエンジン 102 に応じた画像処理、解像度変換等の処理を行う。CMM 130 は、画像データに対して、プロファイルやキャリブレーションデータに基づいた色変換処理（色空間変換処理）を施す専用ハードウェアモジュールである。プロファイルとは、機器に依存した色空間で表現したカラー画像データを、機器に依存しない色空間（例えば Lab 等）に変換するための関数のような情報である。キャリブレーションデータは、スキャナ 101 やプリンタエンジン 102 の色再現特性を修正するためのデータである。

【0017】

図 2 は、本実施形態に係る画像処理装置 100 を制御するソフトウェアモジュールの構成を説明する図である。これらソフトウェアモジュールは、主に CPU 105 の制御の下に動作する。

10

【0018】

ジョブコントロール処理 201 は、各ソフトウェアモジュールを統括・制御し、コピー、プリント、スキャン、FAX 送受信等の画像処理装置 100 で発生するあらゆるジョブの制御を行う。ネットワーク処理 202 は、主にネットワーク I/F 111 を介して行われる、外部装置との通信を制御するモジュールであり、LAN 10 上の各機器との通信制御を行う。ネットワーク処理 202 は、LAN 10 の各機器からの制御コマンドやデータを受信すると、その内容をジョブコントロール処理 201 へ通知する。またジョブコントロール処理 201 からの指示に基づき、LAN 10 の各機器へ制御コマンドやデータの送信を行う。

【0019】

20

UI 処理 203 は、主に操作部 110、操作部 I/F 109 に係る制御を行う。操作者が操作部 110 を操作した内容を、ジョブコントロール処理 201 へ通知すると共に、ジョブコントロール処理 201 からの指示に基づいて、操作部 110 の表示画面の表示内容を制御する。FAX 処理 204 は、モデム 112 を介して FAX 受信を行い、FAX 画像に特有の画像処理を施した後、その受信した画像をジョブコントロール処理 201 へ通知する。また、ジョブコントロール処理 201 からの指定される画像を、指定通知先へ FAX で送信する。プリント処理 207 は、ジョブコントロール処理 201 の指示に基づいて、編集用画像処理部 120、プリンタ画像処理部 119 及びプリンタエンジン 102 を制御し、指定された画像の印刷処理を行う。

【0020】

30

プリント処理 207 は、ジョブコントロール処理 201 より、画像データ、画像情報（画像データのサイズ、カラーモード、解像度等）、レイアウト情報（オフセット、拡大縮小、面つけ等）及び用紙情報（サイズ、印刷方向等）の情報を受付ける。そして画像圧縮部 103、画像伸張部 121、編集用画像処理部 120 及びプリンタ画像処理部 119 を制御して、画像データに対して適切な画像処理を施し、プリンタエンジン 102 を制御して用紙へ印刷を行わせる。

【0021】

スキャン処理 210 は、ジョブコントロール処理 201 の指示に基づいて、スキャナ 101 及びスキャナ画像処理部 118 を制御して、スキャナ 101 により原稿の読み込みを行わせる。ジョブコントロール処理 201 の指示には、カラーモードが含まれており、スキャン処理 210 ではカラーモードに応じた処理が行われる。即ち、カラーモードがカラーであれば原稿をカラー画像として入力し、モノクロであれば原稿をモノクロ画像として入力する。またカラーモードが自動である場合には、プレスキャン等により原稿のカラー/モノクロ判定し、その判定結果に基づいた画像として再度原稿をスキャンして画像を入力する。スキャン処理 210 は、スキャナ 101 により原稿のスキャンを実行し、デジタルデータとして画像データを入力する。こうして入力した画像のカラー情報は、ジョブコントロール処理 201 へ通知される。更に、スキャン処理 210 は、その入力した画像に対し、スキャナ画像処理部 118 を制御して画像の圧縮等の画像処理を施した後、ジョブコントロール処理 201 へ画像処理済みの入力画像を通知する。

40

【0022】

50

色変換処理 209 は、ジョブコントロール処理 201 の指示に基づいて、指示画像に対して色変換処理を行い、色変換処理後の画像をジョブコントロール処理 201 へ通知する。ジョブコントロール処理 201 は、色変換処理 209 に対して、入力色空間情報、出力色空間情報及び色変換を適用する画像を通知する。色変換処理 209 に通知された出力色空間が、入力機器に依存しない色空間（例えば L a b 空間）である場合には、入力機器に依存する入力色空間（例えば、R G B）から L a b に変換するための情報である入力プロファイル情報があわせて通知される。この場合、色変換処理 209 は入力プロファイルより、入力色空間から L a b 空間へマッピングするルックアップテーブル（L U T）を作成し、この L U T を利用して入力画像の色変換を行う。また色変換処理 209 に通知された入力色空間が、L a b 空間である場合には、L a b 空間から出力機器に依存する出力色空間に変換するための出力プロファイル情報があわせて通知される。この場合、色変換処理 209 は出力プロファイルより、L a b 色空間から出力色空間へマッピングする L U T を作成し、この L U T を利用して入力画像の色変換を行う。また、色変換処理 209 に通知された入力色空間、出力色空間の双方が、デバイスに依存する色空間である場合には、入力プロファイルと出力プロファイルの双方が通知される。この場合、色変換処理 209 は入力プロファイル及び出力プロファイルより、入力色空間から出力色空間へダイレクトにマッピングする L U T を作成し、この L U T を利用して入力画像の色変換を行う。色変換処理 209 では、C M M 1 3 0 が機器内であれば、C M M 1 3 0 へ生成した L U T を設定することより、C M M 1 3 0 を利用して色変換を行う。一方、C M M 1 3 0 が無い場合には C P U 1 0 5 がソフト的に色変換処理を行う。尚、色変換処理 209 は、予めリファレンス画像の色変換を実行することにより、入力プロファイルのみが指定された場合の色変換処理に要する時間を保持する。更に、色変換処理 209 は、出力プロファイルのみが指定された場合の色変換に要する時間を保持する。また色変換処理 209 は、入力プロファイルと出力プロファイルが指定された場合の色変換に要する時間を保持する。

【 0 0 2 3 】

R I P 処理 2 1 1 は、ジョブコントロール処理 2 0 1 の指示に基づいて、P D L（ページ記述言語）の解釈（インタプリット）を行い、R I P 部 1 1 6 を制御してレンダリングすることで、ビットマップイメージへの展開を行う。

【 0 0 2 4 】

図 3 は本実施形態に係るプリンタ画像処理部 1 1 9 の構成を示すブロック図である。

【 0 0 2 5 】

タイル分割 D M A C 3 0 0 は、メモリ読出部 3 0 1 とアドレス制御部 3 0 2 を有している。R A M 1 0 6 から原画像データを読み出すときは、アドレス制御部 3 0 2 でアドレスを生成し、そのアドレスに基づいてメモリ読出部 3 0 1 により R A M 1 0 6 へリード要求トランザクションを発行して、R A M 1 0 6 からデータを読み出す。

【 0 0 2 6 】

図 5 は、本実施形態に係る画像処理装置 1 0 0 におけるプリント動作を説明するフローチャートである。この処理は R A M 1 0 6 或いは R O M 1 0 7 に記憶されたプログラムを C P U 1 0 5 が実行することにより達成される。

【 0 0 2 7 】

L A N 1 0 を介して送信されてきた P D L（ページ記述言語）を、ネットワーク I / F 1 1 1 にて受信し、イメージバス I / F 1 1 4 を介して R I P 部 1 1 6 へ入力する。これにより S 1 で、R I P 部 1 1 6 は、その P D L を解釈（インタプリット）してレンダリングを実行する。次に S 2 に進み、S 1 でレンダリングされビットマップ形式の画像データを画像バス 1 1 5 を介して画像圧縮部 1 0 3 に送り、そこで所定の画像圧縮方式を用いて符号化する。そして S 3 で、イメージバス I / F 1 1 4 を介して H D D 1 0 8 へ各ジョブのページ群を格納する。こうして H D D 1 0 8 に格納されたページ群は、S 4 で、プリンタエンジン 1 0 2 の出力タイミングに同期して呼び出され、イメージバス I / F 1 1 4、デバイス I / F 1 2 1 を介して画像伸張部 1 2 1 に送られて復号（伸張）される。こうして復元された画像データを、S 5 で、ページ毎に一旦 R A M 1 0 6 に展開される。こうし

10

20

30

40

50

て展開された画像データは、後述するプリンタ画像処理部 119 のタイル分割 DMA C 300 により、印刷するページのタイミングで読み出され、プリンタ画像処理部 119 で印刷用の画像処理が実行される。S6 では、RAM 106 からタイル画像を読み出し、次に S7 で、そのタイル画像に対して、各フィルタ演算部でタイル単位（ブロック単位）で画像処理を実行する。そして S8 で、全てのタイル画像に対する処理が終了すると S9 に進んで、その処理した画像データをプリンタエンジン 102 に出力して印刷する。

【0028】

図4は、アドレス制御部 302 が生成するアドレスを説明する図である。

【0029】

図は 6 × 6 のタイルデータに周囲 2 画素ずつのオーバーラップタイル画像を付加した 10 × 10 のタイルデータを読み出す場合を示す。図4は展開された画像データを RAM 106 に配列したものである。斜線の部分はオーバーラップ部分（第1の領域）を示し、点線で囲む領域が実際にオーバーラップを含めたタイル部分（ブロック）となる。アドレス制御部 302 は複数のタイルに分割するために、斜線で示したオーバーラップ部を含むタイル部分の画像データを後述するような画素データを読み出せるようにアドレス制御を行う。従って斜線部分で示すオーバーラップ部分の画像データは多重に読み出される。更に、図4のタイル部分の拡大図において、黒丸（ ）が実際に読み出される画像データを示し、その白抜き文字が読み出される順番を示している。即ち、図4に示すように、本身（第2の領域）となる 6 × 6 のタイルデータの全てと、2 × 2 のオーバーラップ部の左上の画素が読み出される。

【0030】

図6は、アドレス制御部 302 がオーバーラップ含むタイル画像を読み出すためのアドレス制御（S6）を示すフローチャートである。

【0031】

まずアドレス制御部 302 は、図4のように列アドレス（X方向）と行アドレス（Y方向）でタイルの位置を判断、制御する。まず始めに S11 で、タイルの左上の画素データのアドレスをスタートタイルアドレスとしてセットする。次に S12 に進み、スタートタイルアドレスでセットされている画素データを読み出す。次に S13 に進み、アドレス制御部 302 が生成しているアドレスが、オーバーラップ部であるか否かを判定する。オーバーラップ部の場合は S14 に進み、更に、行（Y方向）のアドレスを判定し、該当する行に読み出す画素データがあるか否かを判定する。ここで読み出す画素データがない場合は S19 に進んで、行アドレス（Y）を 1 つインクリメントして S12 に進む。一方、読み出す画素データがある場合は S15 に進み、列アドレス（X）を 2 つインクリメントして S17 に進む。S17 では、列の最後か否かを判定し、列の最後でないと判定すると S12 に戻って画素データを読み出す。

【0032】

一方、S13 で、アドレスがオーバーラップ部でないと判定すると S16 に進み、列アドレス（X）を 1 つインクリメントして S17 に進み、画素データを読み出す動作を繰り返す。そして S17 で、列の最後と判定した場合は S18 へ進み、その行がタイルの最終の行であるかどうかを判定する。最終の行でないと判定した場合は S19 に進み行アドレスを 1 つインクリメントして、次の列からの画素を読み出す動作を繰り返す。そして S18 で最終の行であると判定した場合は、タイル画像の読み出しが全て終了したので、この処理を終了する。

【0033】

以上のように本実施形態 1 では、図4のように周囲 2 画素のオーバーラップ部では、2 × 2 の矩形（小ブロック）において左上の画素（代表画素）だけを読み出すようにアドレス制御部 302 のリードアドレスを生成する。これら代表画素は、例えば図7の D0, D1, D2, ... に対応する。これによりオーバーラップ部分の解像度を落とすことができる。またページ全体で並んでいるデータをタイル毎に読み出せるようにアドレスを制御することでタイル画像を生成することができる。これにより図3の入力タイルバッファ 304,

10

20

30

40

50

305には、読み出されたタイルデータが入力されて格納される。セレクタ303は、入力タイルバッファ304、305を切り替えて順次データを格納する。

【0034】

図7は、本実施形態に係る5×5フィルタ演算部307による、注目画素D22に対する演算処理を説明する図である。尚、演算係数CA00～CA08の一例は図9に示されている。

【0035】

本実施形態では、入力タイルバッファ304、305のように2重のバッファ構成にした。これにより、例えば入力タイルバッファ304よりセレクタ306を介して5×5フィルタ演算部307へデータが読み出されている間、他方の入力タイルバッファ305へメモリ読出部301はデータを書き込むことができスループットを高めることができる。5×5フィルタ演算部307は、タイルデータにフィルタ演算を施す演算処理部である。

【0036】

図8は、5×5フィルタ演算部307の内部構造を示すブロック図である。本実施形態では、5×5のフィルタ演算を行う場合の構成を示しており、レジスタ群801と畳み込み演算部802を有している。入力タイルバッファ304、305に格納された画素データは、5×5のフィルタで演算する5×5フィルタ演算部307へ順次入力される。本実施形態では、5×5の垂直方向の5画素をそれぞれ初段のレジスタDin00～Din04の入力へ同時に入力し、この入力動作を5回繰り返して、5×5のフィルタ演算に必要な画素データをレジスタ群内の全てのレジスタに格納する。その後、各レジスタからの出力(Dout00～Dout24)を用いて畳み込み演算部802でフィルタ演算を実行する。

【0037】

図9は、5×5フィルタ演算部307がフィルタ演算に使用するフィルタ係数の一例を示す図である。

【0038】

D22を演算で求める場合は、注目画素に対するフィルタ係数(CA00～CA08)を周辺の対応する画素に乗算する。そして各々の演算結果を用いて5×5のフィルタ画素分の総和を求めて注目画素に対する演算を終了する。

【0039】

図10は、注目画素D6のフィルタ演算を説明する図である。

【0040】

図のように例えば、画素D6の左斜め上の画素は、先ほどのタイル画像読出し時に画素を間引いた代表画素だけを読んでいるために、実際には対象とする画素データがない。この場合は、画素D0を参照画素として演算し、D6の真上、右斜め上の画素は、画素D1を参照する(例えば、参照画素は図の太字のエリアで参照している)。

【0041】

以上のように自身の部分(オーバーラップでない部分)の端部画素の演算を行う場合、参照画素の位置に画素データがない場合には、近傍の画素データを用いて注目画素に対するフィルタ演算を行う。これにより、解像度が低くなっているオーバーラップ部分を参照画素としてフィルタ演算を実行することができる。

【0042】

このようにして自身部分の注目画素のフィルタ演算が終了した画素データは、セレクタ308を介して出力タイルバッファ309或いは310に書き込まれる。出力タイルバッファ309、310のどちらかのバッファが一杯になった時点で、出力タイルの送信が許可状態となり、セレクタ311を介してタイル画像を出力する。

【0043】

これにより図5のS6で、RAM106からタイル画像データを読み出す。S7では、S6で読み出したタイル画像データのフィルタ演算を実行する。そしてS8で、RAM106に展開された画像データの全タイル画像の読出しが終了したか否かを判定し、終了していないときはステップS6に戻るが、終了したと判定した場合はS9に進む。S9では

10

20

30

40

50

、プリンタ画像処理部 119 から出力された画像データをプリンタエンジン 102 へ出力する。これにより画像が用紙に印刷されて処理を終了する。

【0044】

以上説明したように本実施形態 1 によれば、展開されたビットマップ形式の画像データを RAM 106 より読み出す（取得する）際にオーバーラップ部（第 1 の領域）を含み、更に、オーバーラップ部の解像度を低下させて画像データを読み込む。更に、オーバーラップ部が本身（第 2 の領域）よりも相対的に解像度の低いフォーマットでタイル分割を実行する。これにより周辺画素を参照するフィルタ演算を行う場合に、RAM 106 から、オーバーラップ部を含むタイル単位での画像データの転送におけるデータ量を削減できる。従って、RAM 106 のバンドやシステムバス、画像バス等のコスト上昇を抑えて、パフォーマンスの良い画像処理装置を提供できる。

10

【0045】

尚、本実施形態 1 では、プリンタ画像処理部 119 ではフィルタ演算部 307 でフィルタ演算を行ったが、色変換や各種スクリーン処理等を行う処理部を実装し、これら処理部がシリアル、或いは並列に処理する構成でも良い。

【0046】

[実施形態 2]

次に本発明の実施形態 2 を説明する。本実施形態に係る画像処理装置の全体構成及び各ソフトウェアモジュールは前述の実施形態 1 と同様であるため、その説明を省略する。本実施形態 2 では、プリント動作がつのモードを有する場合で説明する。

20

【0047】

図 11 は、ユーザがプリント動作を行う場合のプリンタドライバにおける画面表示の一例を示す図である。この画面は、LAN 10 を介して画像処理装置 100 に接続されている PC（不図示）の表示部に表示される。

【0048】

図 11 は、ユーザが印刷品質を選択するために印刷品質タブ 904 を選択した場合の画面表示例を示す。ここでユーザは、出力方法ダイアログ 905 で、印刷、ストレージへの保存、セキュアプリント等を選択できる。印刷目的ダイアログ 906 は、印刷や保存する際の印刷品質を選択するためのものである。図のように「一般」が選択された場合は、パフォーマンス重視（印刷速度や保存速度が速い）の設定となり、「グラフィックス」、「高精細文書」が選択された場合は画質品質重視の設定となり、これ以降、画像処理装置 100 はこの設定に基づいて動作する。

30

【0049】

次にユーザが印刷目的ダイアログ 906 で「高精細文書」を選択して印刷を指示した場合を図 12 のフローチャートを参照して説明する。

【0050】

図 11 の印刷目的ダイアログ 906 で「高精細文書」を選択し、印刷ボタン 908 を押下すると、LAN 10 に接続された PC より印刷データが PDL（ページ記述言語）で送信される。この PDL データはネットワーク I/F 111 で受信され、イメージパス I/F 114 より RIP 部 116 へ入力される。これにより RIP 部 116 は、S21 で、その PDL データを解釈（インタプリット）してレンダリングを実行する。次に S22 に進み、その展開した画像データを符号化する。そして S23 に進み、イメージパス I/F 114 を介して HDD 108 へ各ジョブのページ群を格納する。次に S24 に進み、格納したページ群を、プリンタエンジン 102 の出力タイミングに同期して読み出して復号して伸張する。次に S25 に進み、こうして復元した画像データを、ページ毎に一旦 RAM 106 に展開する。

40

【0051】

次に S26 に進み、その展開された画像データの印刷品質モードを確認する。ここで高画質モードであると判定すると S27 に進み、後述するプリンタ画像処理部 119 の <RULE 1> に従って画像データを読み出す。一方、S26 で、高画質モードでないと判定

50

するとS 2 8に進み、後述するプリンタ画像処理部 1 1 9の< R U L E 2 >に従って画像データを読み出す。

【 0 0 5 2 】

こうしてS 2 7或いはS 2 8で画像データを読み出すとS 2 9に進み、その画像データに対してフィルタ演算処理を実行する。そしてS 3 0で、全てのタイル画像に対する処理が終了するとS 3 1に進み、プリンタエンジン 1 0 2に出力して印刷する。S 3 0で全てのタイル画像に対する処理が終了していないときはステップS 2 6に戻って、前述の処理を実行する。

【 0 0 5 3 】

図 1 3は、本実施形態 2に係るプリンタ画像処理部 1 1 9の構成を示すブロック図である。本実施形態 2では、実施形態 1で説明した5 × 5フィルタ演算部 3 0 7に、3 × 3フィルタ演算部 1 0 0 7が追加されている。3 × 3フィルタ演算部 1 0 0 7と5 × 5フィルタ演算部 3 0 7の間に緩衝用のタイルバッファ 1 0 0 8が設けられている。その他の構成は図 3と同じであるため同じ記号を付して、その説明を省略する。

10

【 0 0 5 4 】

図 1 4は、本実施形態 2に係るアドレス制御部 1 0 0 2のアドレス制御を説明する図である。前述の実施形態 1とは異なり実施形態 2は、6 × 6のタイルデータに3 × 3のフィルタ演算を行った後、5 × 5のフィルタ演算を行う。従って、実施形態 1のオーバーラップ部の周囲に更に2画素ずつのオーバーラップ部を付加し、本実施形態 2での3 × 3のフィルタ演算を行うために周囲に1画素を付加した計1 2 × 1 2のタイルデータを読み出す。従って、オーバーラップ部は複数種類のサブ領域(ここでは2種類のオーバーラップ部)を含み、各サブ領域はそれぞれ複数の小ブロックに分割されている。

20

【 0 0 5 5 】

図 1 4は、展開された画像データをR A M 1 0 6に配列した状態を示す図である。図において斜線部分はオーバーラップ部分であり、点線で囲まれた領域が実際にオーバーラップ部を含めたタイル部分となる。

【 0 0 5 6 】

アドレス制御部 1 0 0 2は、実施形態 1と同様にタイル画像に分割するために、図の斜線で示したオーバーラップ部を含むタイルを、後述するような順番で画素データを読み出すようにアドレス制御を行う。従って、斜線部分のオーバーラップ部分の画素データは多重に読み出される。更に図 1 4のタイル部分の拡大図において、黒丸()の白抜き文字が実際に読み出す画素データの順番を示している。

30

【 0 0 5 7 】

図 1 5は、実施形態 2に係るアドレス制御部 1 0 0 2がオーバーラップ部分を含むタイル画像の読出しを行うためのアドレス制御を示すフローチャートである。本実施形態 2においても、アドレス制御部 1 0 0 2は、図 1 4のように列アドレス(X方向)と行アドレス(Y方向)の2つのアドレスでタイルの位置を判断、制御する。

【 0 0 5 8 】

まず始めにS 4 1で、画素データのアドレスをスタートタイルアドレスとしてタイルの左上にセットする。次にS 4 2に進み、スタートタイルアドレスでセットされている画素データを読み出す。続いてS 4 3で、アドレス制御部 1 0 0 2が生成している現在のアドレスをチェックし、現在の行アドレスが示す行に読み出す画素データがあるか否かを判断する。ここで読み出す画素データがある場合はS 4 4に進み、次に読み出す画素データをアドレスするように列アドレスを進める。そしてS 4 2に戻って画素データを読み出すという動作を繰り返し、該当の行の読出し対象となる画素データを全てリードする。こうしてS 4 3で、該当する行の画素データを全て読み出した場合はS 4 5に進み、現在の行がタイルの最終行であるか否かを判定する。ここで最終の行でないと判定した場合はS 4 6に進み、行アドレスを1インクリメントしてS 4 3に進み、上記のように次列からの画素を読み出す動作を繰り返す。そしてS 4 5で、最終の行であると判断するとタイル画像の読出し処理を終了する。

40

50

【 0 0 5 9 】

以上説明したように本実施形態 2 では、印刷品質モードを「高精細文書」に設定にした場合は周囲 2 画素のオーバーラップ部分（図の最外周から内側へ 2 周目、3 周目の部分）では、例えば 2 × 2 の矩形の左上の画素）に基づいて読み出す代表画素を選択する。そして図 1 5 で説明したようにアドレスを生成して順次画素データを読み出す（代表画素は例えば図 1 6 の D 6 , D 7 , . . . に対応する）。また 3 × 3 のフィルタ演算用のオーバーラップ部分（図の最外周部分）においても、ある規則（図の場合には 2 × 1 の矩形では左端或いは右端の画素選択、1 × 2 の矩形では上端或いは下端の画素選択）に基づいて代表画素を選択する。これに応じてアドレスを生成し、同じく順次画素データを読み出す（代表画素は例えば図 1 6 の D 0 , D 1 , . . . に対応する）。

10

【 0 0 6 0 】

以上のように例えばフィルタ演算が 2 つある場合、対応するオーバーラップ部分の代表画素をそれぞれ異なるルールでアドレスを飛ばして生成し、読み出すことにより各オーバーラップ部分に応じて異なる解像度をもつタイル分割を行うことができる。また「高精細文書」等の高画質モードが設定された場合は、例えば最外周の読み出す画素の解像度を次段のオーバーラップ部分（最外周から内側へ 2 周目、3 周目の部分）の解像度より高くする等する。こうして、フィルタ演算における画質の向上を実現させ、より高画質な画像の再現を可能とした。

【 0 0 6 1 】

これにより「高精細文書」等の高画質モードにより、アドレス制御部 1 0 0 2 により、上述のルール（図 1 5 では < RULE1 > ）で生成されたアドレスに基づいて、メモリ読み部 1 0 0 1 より R A M 1 0 6 に配列されたデータを読み出す（S 2 7）。こうして読み出されたオーバーラップタイルデータは、セクタ 3 0 3 を介して、入力タイルバッファ 3 0 4 , 3 0 5 のいずれかに格納される。そしてタイルバッファのタイルデータが一杯になると、3 × 3 フィルタ演算部 1 0 0 7 へデータが転送される。この 3 × 3 フィルタ演算部 1 0 0 7 の内部構造は、前述の実施形態 1 のフィルタ演算部 3 0 7 とほぼ同様であるので説明を省略する。

20

【 0 0 6 2 】

同様に 3 × 3 フィルタ演算部 1 0 0 7 の図示しないレジスタ群へ順次データを格納していき、3 × 3 フィルタの計 9 画素がレジスタに格納されると、図示しない畳み込み演算部により 3 × 3 のフィルタ演算が実行される。

30

【 0 0 6 3 】

図 1 6 に示したフィルタ演算式は、図の D 24 を注目画素として演算する場合の例を示している。

【 0 0 6 4 】

図 1 7 は 3 × 3 のフィルタ演算に使用するフィルタ係数の一例を示した図である。

【 0 0 6 5 】

図 1 6 のように、注目画素 D 24 に対する演算の場合は、注目画素に対するフィルタ係数（C A 0 0 ~ C A 0 3）を周辺画素の対応する画素データに乗算する。こうして各々の結果を 3 × 3 のフィルタ画素分の総和をとることで注目画素に対する演算を終了する。

40

【 0 0 6 6 】

一方、図 1 8 のフィルタ演算は、注目画素 D 6 に対する演算の例を示した図である。図のように、例えば注目画素 D 6 の真上の画素には実際には対象とするデータはない。この場合は D 0 を参照画素として演算し、注目画素 D 6 の右横、右斜め下の画素は、自己の D 6 を参照する（例えば、参照画素は図の太字のエリアで参照している）。

【 0 0 6 7 】

以上のように自身の部分（オーバーラップでない部分）の端部画素の演算を行う場合に、参照画素位置にデータがない場合は、近傍の画素データを演算に用いて注目画素に対する演算を行う。これにより解像度が低くなっているオーバーラップ部を参照画素としてフィルタ演算を実行することができる。

50

【 0 0 6 8 】

3 × 3 のフィルタ演算が実行された後、演算結果のタイルデータは、一旦タイルバッファ 1 0 0 8 に格納される。このとき、3 × 3 のフィルタ演算で必要となった最外周のオーバーラップ部分は必要がなくなるので、次段の 5 × 5 フィルタ演算部 3 0 7 へ入力する際にはこの部分を削除する。これにより 1 2 × 1 2 のタイルから 1 0 × 1 0 のタイルへ変更して入力される。こうして 5 × 5 フィルタ演算部 3 0 7 へ入力されたタイルデータには、実施形態 1 と同様の処理が行われる。こうして本身部分の注目画素のフィルタ演算が終了した画素データは、セクタ 3 0 8 を介して出力タイルバッファ 3 0 9 又は 3 1 0 に書き込まれる。そして出力タイルバッファ 3 0 9 , 3 1 0 のいずれかが一杯になった時点でタイルデータの送信が許可状態となり、セクタ 3 1 1 を介してタイルデータが出力される。

10

【 0 0 6 9 】

従って S 2 9 で、プリンタ画像処理部 1 1 9 内の各フィルタ演算部でフィルタ処理を実行しながら、RAM 1 0 6 に展開された画像データの全タイル画像の読出しが終了したか否かを判断する (S 3 0) 。そして全タイル画像の読出しが終了した場合は S 3 1 に進み、プリンタ画像処理部 1 1 9 より出力された画像データをプリンタエンジン 1 0 2 へ入力する。これにより用紙に印刷され、その用紙を排出して印刷動作を終了する。

【 0 0 7 0 】

次にユーザが、図 1 1 の印刷目的ダイアログ 9 0 6 にて「一般」(速度優先モード)に設定して印刷する場合について説明する。RAM 1 0 6 に印刷したい画像データが展開される処理までは「高精細文書」(画質優先モード)が選択された場合と同様なので説明を省略する。

20

【 0 0 7 1 】

今回、図 1 9 のタイル部分の拡大図において、黒丸 () の白抜き文字が実際に読み出す画素データの順番を示している。本モードでのアドレス制御部 1 0 0 2 のアドレスの生成、制御は図 1 5 のフローチャートと同様であるので説明を省略する。

【 0 0 7 2 】

図 1 9 のように、5 × 5 のフィルタ演算用に周囲 2 画素のオーバーラップ部分 (図の最外周から内側へ 2 周目、3 周目の部分) に対しては図のようにある規則 (図の場合は 2 × 2 の矩形において左上の画素を読む) に基づいて読み出す代表画素を選択する。そして、それに基づいてアドレスを生成し、順次読出していく (代表画素は例えば図 2 0 の D 3 , D 4 , . . . に対応する) 。更に今回は、3 × 3 のフィルタ演算用のオーバーラップ部分 (図の最外周部分) においてもある規則 (今回は 4 × 1 の矩形で左 1 、或いは右 1 画素選択、また 1 × 4 の矩形では上端或いは下端の画素選択) に基づいて代表画素を選択する。そして、その選択した画素データを読み出すためにアドレスを生成する (代表画素は例えば図 2 0 の D 0 , D 1 , . . . に対応する) 。

30

【 0 0 7 3 】

以上のように例えばフィルタ演算が 2 つある場合において、対応するオーバーラップ部分の代表画素をそれぞれ異なるルールでアドレスを飛ばして生成し、読み出すことにより各オーバーラップ部分に応じて異なる解像度をもつタイル分割を行うことが可能となる。また「一般」等のパフォーマンス重視モードが設定された場合は、上記のように例えば最外周の読み出す画素の解像度を、「高精細文書」が選択された場合よりも低下させる。具体的には、4 × 1 の矩形で 1 画素を選択している。これにより 3 × 3 のフィルタ演算の精度は劣化する方向だが、タイルデータは小さくなるので RAM 1 0 6 から読み出す際のスループットは早くなりパフォーマンスの向上できる。

40

【 0 0 7 4 】

このように「一般」等の速度優先モードでは、アドレス制御部 1 0 0 2 で、上述したルール (図 1 5 では、< RULE 2 > と記載) で生成されたアドレスに基づいて、メモリ読出部 1 0 0 1 より RAM 1 0 6 から原画像データを読み出す (S 2 8) 。

【 0 0 7 5 】

50

図20, 図21は、3×3フィルタ演算部1007で、それぞれD19とD3のフィルタ演算を行う場合を説明する図である。

【0076】

特に図21において、先ほどの「高精細文書」が選択された場合と比較して、最外周の代表画素が粗くなっているので「FILOUT_03」を求める加算の項は更に重複した参照画素が呼ばれている。従って、参照画素データが少なくなった分だけ演算精度は低くなるが、タイルデータのデータ量を小さくできる。

【0077】

このようにして3×3のフィルタ演算が実行された後、演算結果のタイルデータは一旦タイルバッファ1008に格納される。前述と同様に3×3のフィルタ演算で必要となった最外周のオーバーラップ部分は必要がなくなるので、次段の5×5フィルタ演算部307に入力する際にはこの部分を削除して、12×12のタイルから10×10のタイルへ変更される。次段の5×5フィルタ演算部307へ入力されたタイルデータは、前述の実施形態1と同様の処理が行われ、本身部分の注目画素のフィルタ演算が終了した画素はセクタ308を介して、出力タイルバッファ309, 310に書き込まれる。そして出力タイルバッファ309, 310のいずれかが一杯になるとタイルデータの送信が許可状態となりセクタ311を介してタイルデータを出力する(S29)。従って、S29でプリンタ画像処理部119の各フィルタ演算部でフィルタ処理を実行しながら、RAM106に展開された画像データの全タイル画像の読出しが終了したか否かを判定する(S30)。全タイル画像の読出しが終了した場合はS31へ進んで、プリンタエンジン102へ出力して印刷してプリント動作を終了する。

【0078】

以上のように本実施形態2では、ユーザが設定した印刷モード(画像処理モード)に応じて、ページ画像を展開したRAM等のページメモリよりデータを読み出す際にオーバーラップ部分の解像度を切替えて画像データを読み出すように制御できる。従って、オーバーラップ部分を含むタイル画像のデータ量を、その印刷モードに応じて変更することができ、これにより画像データの増大を抑えて、ユーザが意図した画像品質と処理速度で印刷できる。

【0079】

また本実施形態2のように、プリンタ画像処理部119に周辺画素を参照するフィルタ演算等の処理が複数ある場合には、フィルタ演算用のオーバーラップ部を対応する複数の階層とすることで複数の異なるフィルタ演算処理にも対応できるようになる。更に画像処理部が、印刷モード等に応じて、読み出す画素データの解像度を変更することにより、RAMとの間でデータ帯域を広げることなく、ユーザのモード設定等に適したパフォーマンスの良い処理を実行できる。

【0080】

[実施形態3]

前述の実施形態1, 2では、ページ単位で画像データを圧縮していた。このため、任意のタイルを切り出すためには、1ページ全部の画像データをRAM106に展開する必要がある。例えば、A4縦の画像データをA4横で印刷するために90度或いは270度回転する場合、或いは回転を伴わない場合でも、例えば10×10のタイルを切り出すためには(9ライン+)の画像を伸張する必要がある。そのため、画像データを伸張するために使用するRAM106の容量が大きくなってしまふ。

【0081】

そこで本実施形態3では、図5のS2の画像データの圧縮の段階でオーバーラップ部分を含むタイルに分割してからJPEG圧縮する。その際に、JPEGの処理単位の都合上、本身の部分とオーバーラップ部分共に、そのサイズは縦、横ともに8の倍数とする。この制限の下で、本身の部分とオーバーラップ部分のサイズは、後段のプリンタ画像処理部119での処理(S7)で必要なサイズによって決める。この方法によって、図12の画像伸張(S25)の際に、1ページの中の必要な部分のみをRAM106に展開すること

10

20

30

40

50

ができ、その際に必要なRAM106の容量を小さくすることができる。

【0082】

本実施形態3では、オーバーラップ部分は、複数のオーバーラップ部分を含むタイルに含まれるため、普通に圧縮したのでは圧縮後のデータが大きくなってしまふ。そこで、本実施形態3では、オーバーラップ部分と自身の部分とで圧縮パラメータを切り替える。それによって、最終的な印刷品質への影響を抑えつつデータ量を小さくしている。

【0083】

図22(A)は、本実施形態3に係る画像圧縮部103の構成を示すブロック図である。図23は、本実施形態3に係る画像圧縮部103による圧縮処理を示すフローチャートである。これらを参照しながら圧縮方法について説明する。この処理は、図5のS2或いは図12のS22で実行される。

10

【0084】

MCU DMAC(以下、DMAC)2401は、指定された自身の部分とオーバーラップ部分のサイズをもとにRAM106のビットマップデータからオーバーラップ部分を含むタイルを切り出して読み出す(S51)。JPEGでは図24(A)~(C)に示すようにMCU(Minimum Code Unit)と呼ばれる単位で処理される。MCUは8×8画素で構成され、各コンポーネント(R,G,B,C,M,Y,K等)ごとに存在する。図24(A)~(C)のそれぞれは、R,G,Bのそれぞれに対するMCUを示している。

【0085】

次にS52に進み、DMAC2401は、読み出したビットマップイメージをMCU単位に並び替える。その際に、DMAC2401は、各MCUがオーバーラップ部分であるか、自身の部分であるかを識別し、それに基づく量子化テーブルの選択信号2805(図26)を生成して後段のモジュールに伝播していく。そしてS53で、各MCUは、DCT部2402によって離散コサイン変換されて2次元の周波数成分に変換される。

20

【0086】

図25(A)に示すAfterDCTは、MCUを離散コサイン変換した場合の一例を示し、変換後も8×8単位である。各要素は周波数変換後の係数値を示し、一番左上の値はDC成分の係数値を示し、右下に行くほど高調波成分の係数値を示す。こうして周波数変換された各MCUは次に、量子化部2403によって量子化される。

【0087】

図26は、本実施形態3に係る量子化部2403の構成を示すブロック図である。

30

【0088】

この量子化部2403は、量子化テーブルを参照して量子化を行う量子化コア部2801、選択信号2805に応じて、量子化に使用する量子化テーブルを選択するセレクタ2802を有している。量子化テーブル2803,2804はそれぞれ、オーバーラップしたMCU用の量子化テーブルと、オーバーラップしていない(自身の)MCU用の量子化テーブル(図27(B))である。ここで選択信号2805が「0」であれば量子化テーブル2803(図27(A))が、「1」であれば量子化テーブル2804(図27(B))がそれぞれ選択される。

【0089】

この量子化コア部2801は、MCUの各要素をそれぞれに対応する量子化値で割り算して量子化する。8×8のMCUに対応する量子化値をまとめて表記したものを量子化テーブルと呼ぶ。この量子化テーブルの例を図27(A)(B)に示す。一般的に、高調波成分の情報は失われても人間の視覚では目立たないことが知られているため、高調波成分に対応する量子化値ほど大きい(粗い)値を使用する。オーバーラップしたMCU用のテーブルである図27(A)に示す例で説明すると、右下に行くほど大きい値になる。図27(B)は、自身のMCU用の量子化テーブル2804の具体例を示している。

40

【0090】

JPEG圧縮では、後に出てくる符号化部2404のハフマン符号化の性質上、量子化テーブルの値を大きくしたほうが圧縮後のコードデータのサイズを小さくできる。但し、

50

量子化テーブルの値を大きくしていくと失われる情報量も多くなって人間の視覚でも劣化が目立つようになる。つまり画像品質とコードデータのサイズはトレードオフであり、それを決める要因が量子化テーブルの値である。

【 0 0 9 1 】

本実施形態3では、オーバーラップ部分は自身の部分に比べて、後段の画像処理を行った後の最終画像データに対する影響度が小さいことに着目し、オーバーラップ部分と自身の部分で量子化テーブルの値を変えている。それによって、自身の部分は高画質に、オーバーラップ部分は自身と比べると画質は落ちるが、圧縮後のデータサイズが小さくなるようにする。

【 0 0 9 2 】

具体的には、S54でオーバーラップ部分かどうかを判定し、セレクタ2802に選択信号2805を出力する。量子化コア部2801は、DMAC2410から受け取った周波数変換後の係数値を、セレクタ2802で選択された対応する量子化テーブルの値で割り算する。

【 0 0 9 3 】

図25(A)は、周波数変換後の係数値を示す。図25(B)は、量子化テーブル2803で量子化した(S55)結果の一例を示し、図25(C)は、量子化テーブル2804を使用して量子化した(S56)結果の一例を示す。尚、ここでは、割り算の結果を切り捨てているが、四捨五入する方法でも良い。こうしてS55或いはS56を実行した後S57に進み、その量子化した値をハフマン符号化する。

【 0 0 9 4 】

符号化部2404によるハフマン符号化では、出現頻度の高い数値に対して短いビット長を割り当てることによりデータ量を削減する。量子化後の値では「0」が一番出現頻度が高いはずである。よって、「0」に短いビット長を割り当てる。つまり、JPEG圧縮では、量子化テーブルの値を調整して画質に影響の少ない高調波成分の係数値をできるだけ「0」にすることにより、できる限り画質を維持したままデータ量を削減する。次にS58に進み、ハフマン符号化されたコードデータをCode DMAc2405によりRAM106に出力して保存し、画像圧縮処理を終了する。

【 0 0 9 5 】

次に図5のS4、図12のS24の画像伸張処理について説明する。この画像伸張処理は、前述の画像圧縮処理のほぼ逆順に処理を行う。

【 0 0 9 6 】

図22(B)は、本実施形態3に係る画像伸張部121の構成を示すブロック図である。図28は、本実施形態3に係る画像伸張部121の処理を説明するフローチャートである。

【 0 0 9 7 】

図29は、画像伸張部121の逆量子化部2503の構成を示すブロック図である。

【 0 0 9 8 】

S61で、Code DMAc2501がRAM106からハフマン符号化されたコードデータを読み出す。次にS62に進み、ハフマン復号化部2502がハフマン復号化を実行し、S63で、その復号化したデータをMCU単位に並べる。つぎに逆量子化部2503で逆量子化を行う。この逆量子化では、量子化の際に使用したのと同じ量子化テーブルを使用する。具体的には、まずS64で、オーバーラップ部分かどうかを判定する。そしてその判定した結果を信号3204でセレクタ3202に伝える。ここで信号3204が「0」であれば、セレクタ3202はオーバーラップ部分用テーブル3203を選択し、「1」であれば自身部分用テーブル3204を選択する。これにより逆量子化コア部3201は、その選択されたテーブルを使用して逆量子化を行う。具体的には、逆量子化部コア部3201は、ハフマン復号部2502から入力した量子化値を、その選択された量子化テーブルの値で掛け算して量子化する(S65, S66)。

【 0 0 9 9 】

図30(A)(B)は逆量子化後の値の一例を示す図で、図30(A)は、S65で逆量子化した結果例を示し、図30(B)は、S66で逆量子化した結果例を示す。

【0100】

次にS67に進み、その逆量子化したデータを、逆DCT(逆離散コサイン変換)部2504で逆離散コサイン変換する。次にS68に進み、ラスタDMAC2505によってMCU単位のデータをビットマップデータに変換する。そしてS69に進み、そのラスタデータをRAM106に保存して、この画像伸張処理を終了する。

【0101】

(他の実施形態)

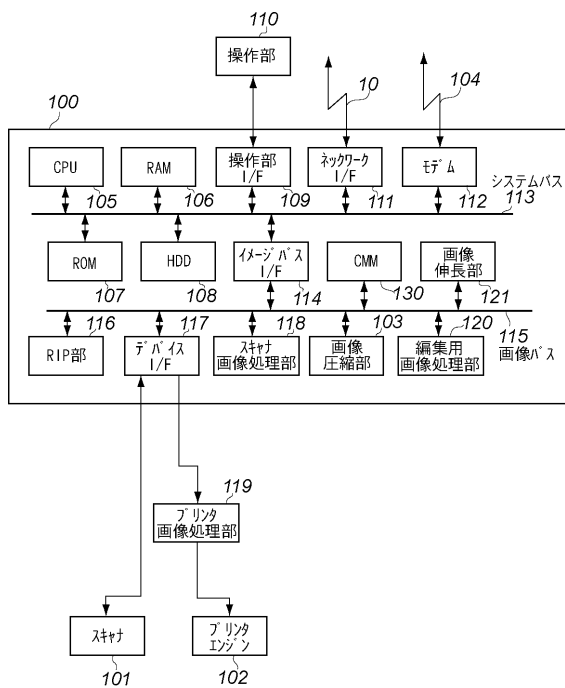
また、本発明の目的は、以下の処理を実行することによっても達成される。即ち、上述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システム或いは装置に供給し、そのシステム或いは装置のコンピュータ(またはCPUやMPU等)が記憶媒体に格納されたプログラムコードを読み出す処理である。

10

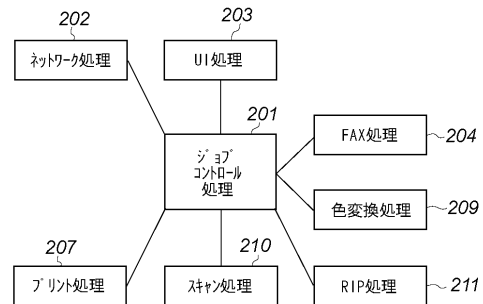
【0102】

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の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及び該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。

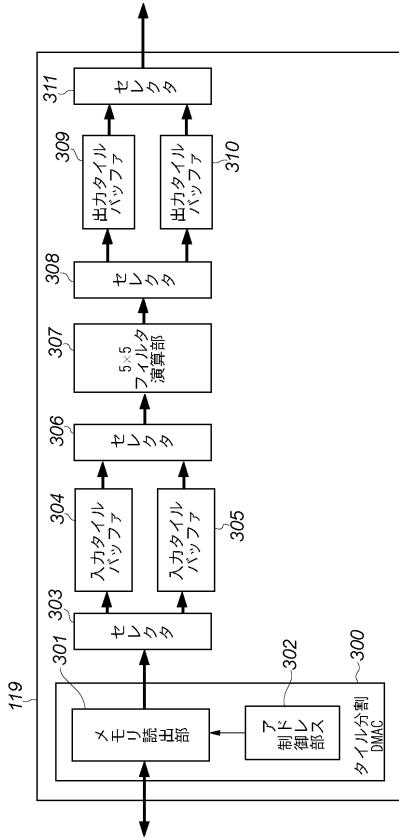
【図1】



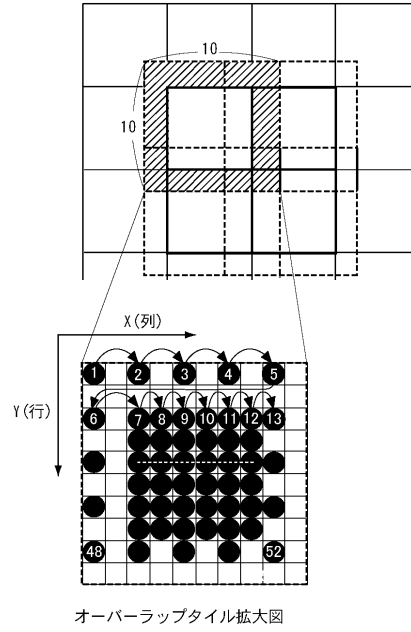
【図2】



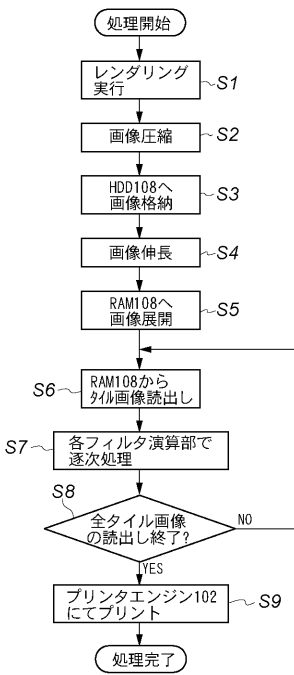
【図3】



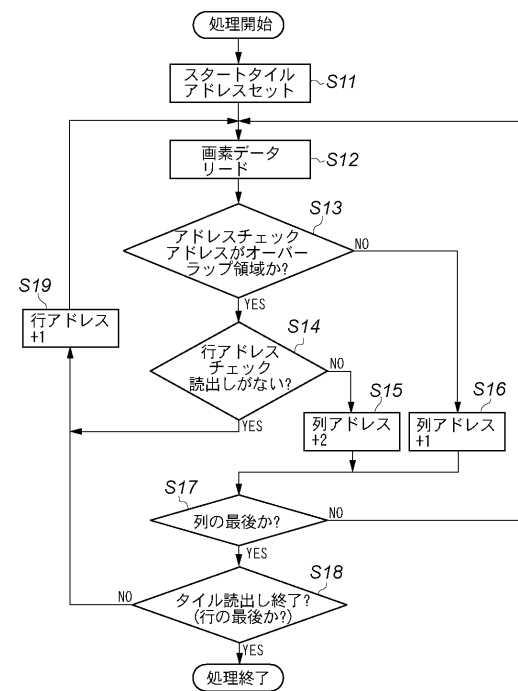
【図4】



【図5】

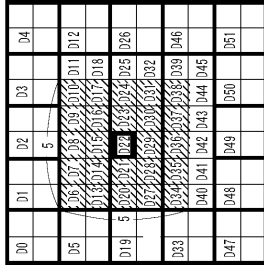


【図6】

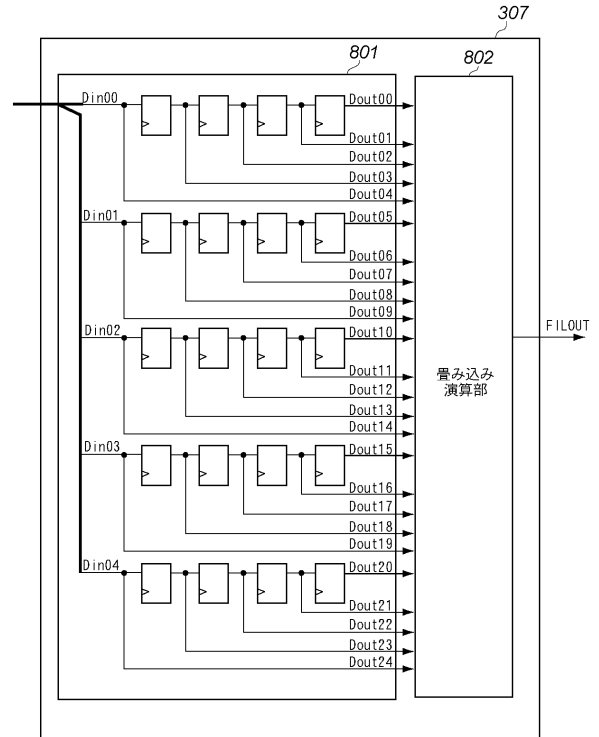


【図7】

フィルタ演算式
 F1L0UT_22= CA00×D22
 +CA01×(D15+D29)
 +CA02×(D14+D16+D28+D30)
 +CA03×(D21+D23)
 +CA04×(D8+D36)
 +CA05×(D7+D9+D35+D37)
 +CA06×(D6+D10+D34+D38)
 +CA07×(D13+D17+D27+D31)
 +CA08×(D20+D24)



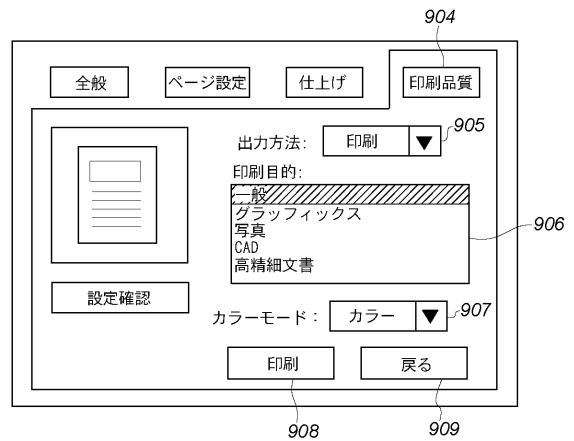
【図8】



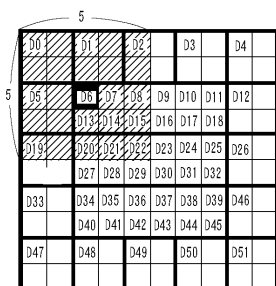
【図9】

CA06	CA05	CA04	CA05	CA06
CA07	CA02	CA01	CA02	CA07
CA08	CA03	CA00	CA03	CA08
CA07	CA02	CA01	CA02	CA07
CA06	CA05	CA04	CA05	CA06

【図11】

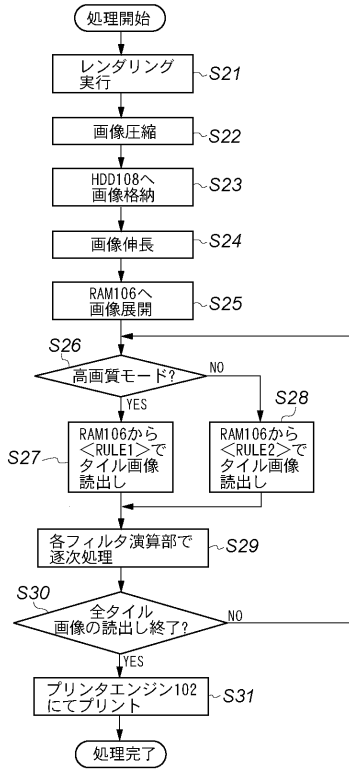


【図10】

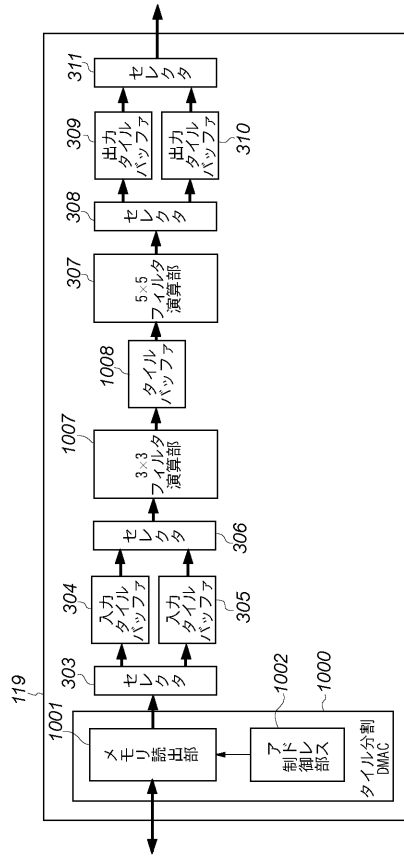


フィルタ演算式
 F1L0UT_06= CA00×D6
 +CA01×(D1+D13)
 +CA02×(D0+D1+D5+D14)
 +CA03×(D5+D7)
 +CA04×(D1+D20)
 +CA05×(D0+D1+D19+D21)
 +CA06×(D0+D2+D19+D22)
 +CA07×(D0+D2+D5+D15)
 +CA08×(D5+D8)

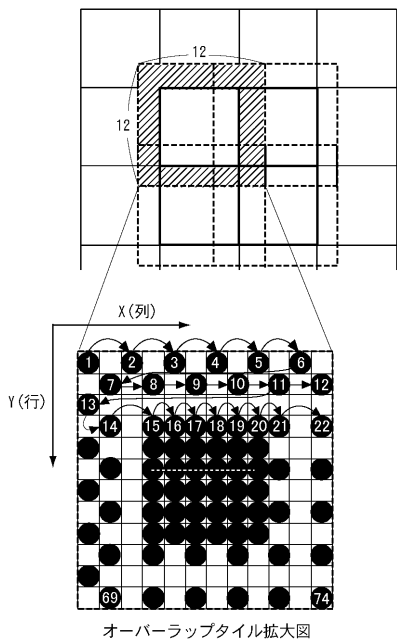
【図12】



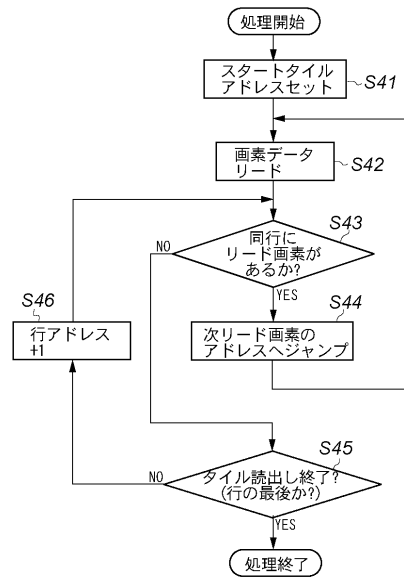
【図13】



【図14】



【図15】



【図 16】

フィルタ演算式

$$\text{FILOUT}_{24} = \text{CA00} \times \text{D24} + \text{CA01} \times (\text{D15} + \text{D31}) + \text{CA02} \times (\text{D14} + \text{D16} + \text{D30} + \text{D32}) + \text{CA03} \times (\text{D23} + \text{D25})$$

D0	D1	D2	D3	D4	D5
D6	D7	D8	D9	D10	D11
D12	D13	D14	D15	D16	D17
D18	D19	D20	D21	D22	D23
D24	D25	D26	D27	D28	D29
D30	D31	D32	D33	D34	D35
D36	D37	D38	D39	D40	D41
D42	D43	D44	D45	D46	D47
D48	D49	D50	D51	D52	D53
D54	D55	D56	D57	D58	D59
D60	D61	D62	D63	D64	D65
D66	D67	D68	D69	D70	D71
D72	D73				

【図 17】

CA05	CA01	CA02
CA03	CA00	CA03
CA02	CA01	CA02

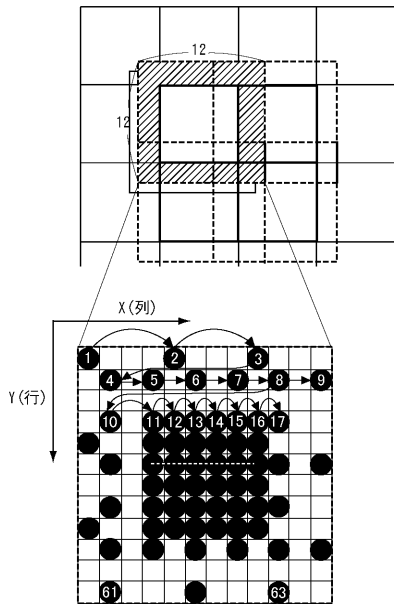
【図 18】

D0	D1	D2	D3	D4	D5
D6	D7	D8	D9	D10	D11
D12	D13	D14	D15	D16	D17
D18	D19	D20	D21	D22	D23
D24	D25	D26	D27	D28	D29
D30	D31	D32	D33	D34	D35
D36	D37	D38	D39	D40	D41
D42	D43	D44	D45	D46	D47
D48	D49	D50	D51	D52	D53
D54	D55	D56	D57	D58	D59
D60	D61	D62	D63	D64	D65
D66	D67	D68	D69	D70	D71
D72	D73				

フィルタ演算式

$$\text{FILOUT}_{06} = \text{CA00} \times \text{D6} + \text{CA01} \times (\text{D0} + \text{D6}) + \text{CA02} \times (\text{D0} + \text{D1} + \text{D12} + \text{D6}) + \text{CA03} \times (\text{D12} + \text{D6})$$

【図 19】



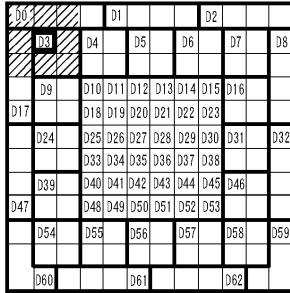
【図 20】

フィルタ演算式

$$\text{FILOUT}_{19} = \text{CA00} \times \text{D19} + \text{CA01} \times (\text{D11} + \text{D26}) + \text{CA02} \times (\text{D10} + \text{D12} + \text{D25} + \text{D27}) + \text{CA03} \times (\text{D18} + \text{D20})$$

D0	D1	D2	D3	D4	D5
D6	D7	D8	D9	D10	D11
D12	D13	D14	D15	D16	D17
D18	D19	D20	D21	D22	D23
D24	D25	D26	D27	D28	D29
D30	D31	D32	D33	D34	D35
D36	D37	D38	D39	D40	D41
D42	D43	D44	D45	D46	D47
D48	D49	D50	D51	D52	D53
D54	D55	D56	D57	D58	D59
D60	D61	D62	D63	D64	D65
D66	D67	D68	D69	D70	D71
D72	D73				

【図 2 1】



フィルタ演算式

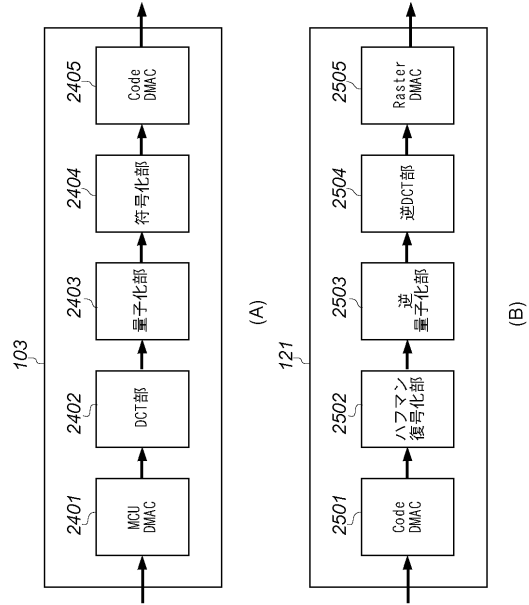
$$FILOUT_03 = CA00 \times D3$$

$$+ CA01 \times (D0 + D3)$$

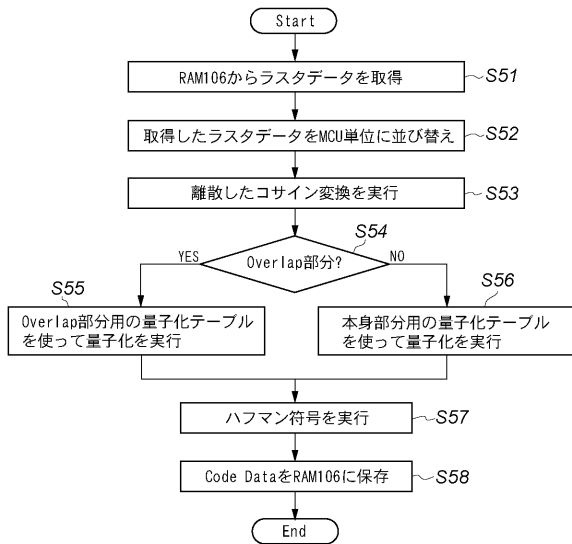
$$+ CA02 \times (D0 + D0 + D17 + D3)$$

$$+ CA03 \times (D17 + D3)$$

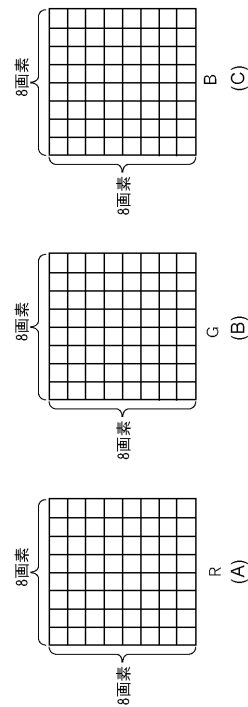
【図 2 2】



【図 2 3】



【図 2 4】



【図 25】

(0,0) AfterDCT

56	87	86	97	86	75	75
86	85	87	100	54	100	57
86	86	65	46	95	97	78
64	73	53	57	75	87	67
64	53	35	78	78	46	16
31	32	42	23	13	32	8
89	15	24	56	42	8	8
46	77	64	12	14	13	15

AfterDCT (7,7) (A)

(0,0) QT_A

14	8	4	3	0	0	0
8	4	3	2	0	0	0
4	2	1	0	0	0	0
2	1	0	0	0	0	0
1	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

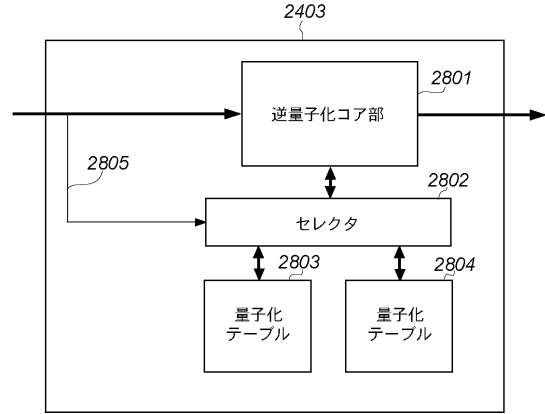
QT_A(i, j)=AfterDCT(i, j)/QT_A (7,7) (B)

(0,0) QT_B

14	4	4	4	3	4	3
4	4	4	5	2	5	2
4	4	3	2	4	4	3
3	3	2	2	3	4	3
3	2	1	3	3	2	0
1	1	2	1	0	1	0
4	0	1	2	2	0	0
2	3	3	0	0	0	0

QT_B(i, j)=AfterDCT(i, j)/QT_B (7,7) (C)

【図 26】



【図 27】

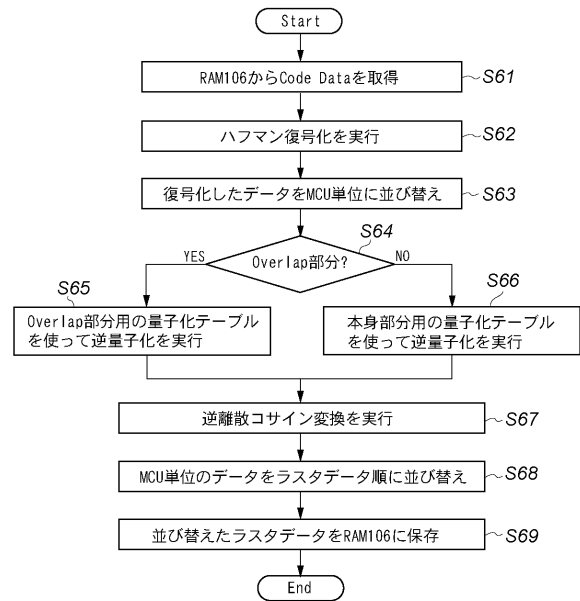
Table B (For substance MCU) (B)

20	20	20	20	20	20	20	20	20	20
20	20	20	20	20	20	20	20	20	20
20	20	20	20	20	20	20	20	20	20
20	20	20	20	20	20	20	20	20	20
15	20	20	20	20	20	20	20	20	20
10	15	20	20	20	20	20	20	20	20
4	10	15	20	20	20	20	20	20	20

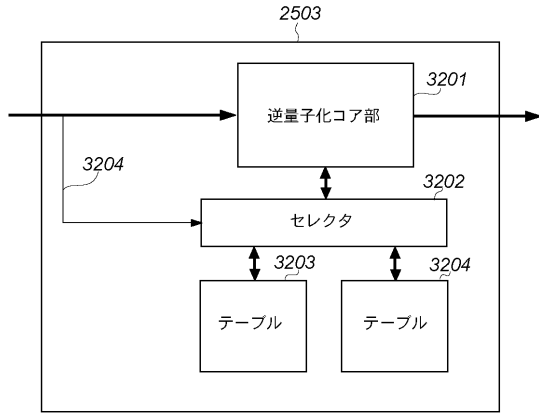
Table A (For overlap MCU) (A)

4	10	20	30	40	100	100	100	100	100
10	20	30	40	100	100	100	100	100	100
20	30	40	100	100	100	100	100	100	100
30	40	100	100	100	100	100	100	100	100
40	100	100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100	100

【図 28】



【図 29】



【図 30】

(A)

56	80	80	90	0	0	0	0	0	0
80	80	90	80	0	0	0	0	0	0
80	60	40	0	0	0	0	0	0	0
60	40	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$Inv_QT_A(i, j) = QT_A(i, j) * QT_TableA$

(B)

56	40	60	80	80	60	60	60	60	60
40	60	80	100	40	100	40	40	40	40
60	80	80	40	80	80	80	80	60	60
60	60	40	40	60	80	40	60	60	40
60	40	20	60	60	40	0	40	0	40
20	20	40	20	0	20	0	0	0	0
80	0	20	40	40	40	0	0	0	0
40	60	60	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$Inv_QT_B(i, j) = QT_B(i, j) * QT_TableB$

フロントページの続き

- (72)発明者 山田 直人
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 西山 淳
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 伊藤 隆夫

- (56)参考文献 特開2005-198121(JP,A)
特開平01-256888(JP,A)

- (58)調査した分野(Int.Cl., DB名)
- | | |
|------|-------------|
| G06T | 1/00 - 5/50 |
| H04N | 1/40 |