

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号
特許第4408836号
(P4408836)

(45) 発行日 平成22年2月3日 (2010.2.3)

(24) 登録日 平成21年11月20日 (2009.11.20)

(51) Int.Cl.	F I
HO 4 N 1/00 (2006.01)	HO 4 N 1/00 C
HO 4 N 1/41 (2006.01)	HO 4 N 1/41 Z
B 4 1 J 29/38 (2006.01)	B 4 1 J 29/38 Z
GO 6 F 3/12 (2006.01)	GO 6 F 3/12 B

請求項の数 9 (全 53 頁)

(21) 出願番号	特願2005-157607 (P2005-157607)	(73) 特許権者	000001007
(22) 出願日	平成17年5月30日 (2005.5.30)		キヤノン株式会社
(65) 公開番号	特開2006-333361 (P2006-333361A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成18年12月7日 (2006.12.7)	(74) 代理人	100076428
審査請求日	平成20年5月29日 (2008.5.29)		弁理士 大塚 康徳
		(74) 代理人	100112508
			弁理士 高柳 司郎
		(74) 代理人	100115071
			弁理士 大塚 康弘
		(74) 代理人	100116894
			弁理士 木村 秀二
		(72) 発明者	木虎 正和
			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
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、プログラム

(57) 【特許請求の範囲】

【請求項 1】

入力された画像データに対する処理を実行する画像処理装置であって、
画像データを入力する入力手段と、
画像データを出力する出力手段と、
ラスト画像データを、所定の大きさのブロック毎に分割し、ベクトル化処理を実行することにより、当該所定の大きさに分割された各ブロックに対応するブロックベクタ画像データに変換する第1変換手段と、
ブロックベクタ画像データを記憶する記憶手段と、
ブロックベクタ画像データをラスト画像データに展開する展開手段と、
前記入力手段から入力されたラスト画像データを前記第1変換手段でブロックベクタ画像データに変換させ、当該変換されたブロックベクタ画像データを前記記憶手段に記憶させ、前記記憶手段に記憶されているブロックベクタ画像データを前記展開手段で展開して得られるラスト画像データを前記出力手段から出力させるように、装置内での処理対象の画像データの転送を制御する画像データ転送制御手段と、
1ページ分のブロックベクタ画像データから、ページ全体を示すページベクタ画像データへ変換する第3変換手段とを備え、
前記展開手段は、複数のブロックベクタ画像データを並列に処理するための、複数の小展開部を有し、
前記画像データ転送制御手段は、

10

20

前記出力手段から出力させるべき画像データの形式がラスタ画像データ形式であると判断した場合は、前記記憶手段に記憶されたブロックベクタ画像データに対して前記展開手段を実行させて得られるラスタ画像データを前記出力手段から出力させ、

前記出力手段から出力させるべき画像データの形式がベクタ画像データ形式であると判断した場合は、前記記憶手段に記憶されたブロックベクタ画像データに対して前記第3変換手段を実行させて得られる前記ページベクタ画像データを前記出力手段から出力させる

ように、装置内での処理対象の画像データの転送を制御することを特徴とする画像処理装置。

【請求項2】

入力された画像データに対する処理を実行する画像処理装置であって、
画像データを入力する入力手段と、
画像データを出力する出力手段と、
ラスタ画像データを、所定の大きさのブロック毎に分割し、ベクトル化処理を実行することにより、当該所定の大きさに分割された各ブロックに対応するブロックベクタ画像データに変換する第1変換手段と、

ブロックベクタ画像データを記憶する記憶手段と、

ブロックベクタ画像データをラスタ画像データに展開する展開手段と、

前記入力手段から入力されたラスタ画像データを前記第1変換手段でブロックベクタ画像データに変換させ、当該変換されたブロックベクタ画像データを前記記憶手段に記憶させ、前記記憶手段に記憶されているブロックベクタ画像データを前記展開手段で展開して得られるラスタ画像データを前記出力手段から出力させるように、装置内での処理対象の画像データの転送を制御する画像データ転送制御手段とを備え、

前記展開手段は、複数のブロックベクタ画像データを並列に処理するための、複数の小展開部と、前記複数の小展開部に対して共通のフォントキャッシュ部とを備え、

前記フォントキャッシュ部は、フォントキャッシュメモリと、前記複数の小展開部それぞれが前記フォントキャッシュメモリを参照中であることを示すロックフラグを記憶するロックフラグレジスタを備える

ことを特徴とする画像処理装置。

【請求項3】

前記展開手段は、前記小展開部での処理対象のブロックベクタ画像データに含まれるフォントデータに対応する展開済フォントデータが前記フォントキャッシュ部に存在する場合、対応するロックフラグレジスタの前記ロックフラグをセットし、前記展開済フォントデータの読出が終了した時点で、前記ロックフラグをクリアする

ことを特徴とする請求項2に記載の画像処理装置。

【請求項4】

前記展開手段は、前記小展開部で展開対象のフォントデータをラスタデータに展開する際に、前記ラスタデータを展開するための空き領域が前記フォントキャッシュメモリに存在しない場合は、前記ロックフラグレジスタでロックフラグがセットされていない該フォントキャッシュメモリ上の展開済フォントデータを、前記展開対象のフォントデータに置き換える

ことを特徴とする請求項3に記載の画像処理装置。

【請求項5】

前記複数の小展開部それぞれに、対応するロックフラグレジスタを制御するための制御信号線が接続されている

ことを特徴とする請求項2に記載の画像処理装置。

【請求項6】

入力された画像データに対する処理を実行する画像処理装置の制御方法であって、
ラスタ画像データを、所定の大きさのブロック毎の分割とベクトル化処理とにより、所定の大きさに分割されたブロックベクタ画像データに変換する第1変換工程と、

ブロックベクタ画像データを記憶手段に記憶する記憶工程と、

ブロックベクタ画像データをラスタ画像データに展開する展開工程と、

前記入力されたラスタ画像データを前記第1変換工程でブロックベクタ画像データに変換させ、当該変換されたブロックベクタ画像データを前記記憶手段に記憶させ、前記記憶手段に記憶されているブロックベクタ画像データを前記展開工程で展開して得られるラスタ画像データを出力部から出力させるように、装置内での処理対象の画像データの転送を制御する画像データ転送制御工程と、

1 ページ分のブロックベクタ画像データから、ページ全体を示すページベクタ画像データへ変換する第3変換工程とを備え、

前記展開工程は、複数の小展開部によって、複数のブロックベクタ画像データを並列に処理し、

前記画像データ転送制御工程は、

前記出力部から出力させるべき画像データの形式がラスタ画像データ形式であると判断した場合は、前記記憶手段に記憶されたブロックベクタ画像データに対して前記展開工程を実行させて得られるラスタ画像データを前記出力部から出力させ、

前記出力部から出力させるべき画像データの形式がベクタ画像データ形式であると判断した場合は、前記記憶手段に記憶されたブロックベクタ画像データに対して前記第3変換工程を実行させて得られる前記ページベクタ画像データを前記出力部から出力させる

ように、装置内での処理対象の画像データの転送を制御する

ことを特徴とする画像処理装置の制御方法。

【請求項7】

入力された画像データに対する処理を実行する画像処理装置の制御方法であって、

ラスタ画像データを、所定の大きさのブロック毎の分割とベクトル化処理とにより、所定の大きさに分割されたブロックベクタ画像データに変換する第1変換工程と、

ブロックベクタ画像データを記憶手段に記憶する記憶工程と、

ブロックベクタ画像データをラスタ画像データに展開する展開工程と、

前記入力されたラスタ画像データを前記第1変換工程でブロックベクタ画像データに変換させ、当該変換されたブロックベクタ画像データを前記記憶手段に記憶させ、前記記憶手段に記憶されているブロックベクタ画像データを前記展開工程で展開して得られるラスタ画像データを出力部から出力させるように、装置内での処理対象の画像データの転送を制御する画像データ転送制御工程とを備え、

前記展開工程は、複数の小展開部と該複数の小展開部に対する共通のフォントキャッシュ部とによって、複数のブロックベクタ画像データを並列に処理し、

前記フォントキャッシュ部は、フォントキャッシュメモリと、前記複数の小展開部それぞれが前記フォントキャッシュメモリを参照中であることを示すロックフラグを記憶するロックフラグレジスタを備える

ことを特徴とする画像処理装置の制御方法。

【請求項8】

入力された画像データに対する処理をコンピュータに実行させる画像処理装置の制御を実現するプログラムであって、

ラスタ画像データを、所定の大きさのブロック毎の分割とベクトル化処理とにより、所定の大きさに分割されたブロックベクタ画像データに変換する第1変換工程と、

ブロックベクタ画像データを記憶手段に記憶する記憶工程と、

ブロックベクタ画像データをラスタ画像データに展開する展開工程と、

前記入力されたラスタ画像データを前記第1変換工程でブロックベクタ画像データに変換させ、当該変換されたブロックベクタ画像データを前記記憶手段に記憶させ、前記記憶手段に記憶されているブロックベクタ画像データを前記展開工程で展開して得られるラスタ画像データを出力部から出力させるように制御する制御工程と、

1 ページ分のブロックベクタ画像データから、ページ全体を示すページベクタ画像データへ変換する第3変換工程とを備え、

前記展開工程は、複数の小展開部によって、複数のブロックベクタ画像データを並列に処理し、

前記制御工程は、

前記出力部から出力させるべき画像データの形式がラスタ画像データ形式であると判断した場合は、前記記憶手段に記憶されたブロックベクタ画像データに対して前記展開工程を実行させて得られるラスタ画像データを前記出力部から出力させ、

前記出力部から出力させるべき画像データの形式がベクタ画像データ形式であると判断した場合は、前記記憶手段に記憶されたブロックベクタ画像データに対して前記第3変換工程を実行させて得られる前記ページベクタ画像データを前記出力部から出力させる

ように、装置内での処理対象の画像データの転送を制御する

10

ことをコンピュータに実行させることを特徴とするプログラム。

【請求項9】

入力された画像データに対する処理をコンピュータに実行させる画像処理装置の制御を実現するプログラムであって、

ラスタ画像データを、所定の大きさのブロック毎の分割とベクトル化処理とにより、所定の大きさに分割されたブロックベクタ画像データに変換する第1変換工程と、

ブロックベクタ画像データを記憶手段に記憶する記憶工程と、

ブロックベクタ画像データをラスタ画像データに展開する展開工程と、

前記入力されたラスタ画像データを前記第1変換工程でブロックベクタ画像データに変換させ、当該変換されたブロックベクタ画像データを前記記憶手段に記憶させ、前記記憶手段に記憶されているブロックベクタ画像データを前記展開工程で展開して得られるラスタ画像データを出力部から出力させるように制御する制御工程とを備え、

20

前記展開工程は、複数の小展開部と該複数の小展開部に対する共通のフォントキャッシュ部とによって、複数のブロックベクタ画像データを並列に処理し、

前記フォントキャッシュ部は、フォントキャッシュメモリと、前記複数の小展開部それぞれが前記フォントキャッシュメモリを参照中であることを示すロックフラグを記憶するロックフラグレジスタを備える

ことをコンピュータに実行させ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】

30

【0001】

本発明は、入力された画像データに対して画像処理を施して管理する画像処理装置及びその制御方法、プログラムに関するものである。

【背景技術】

【0002】

ネットワーク上に接続され、外部あるいは内部の画像データに対する画像データ処理を実行し、その処理した画像データを出力する画像データ入出力システムが知られている。

【0003】

この画像データ入出力システムとしては、MFP(Multi Function Peripheral)と呼ばれるものがある。

40

【0004】

ここで、従来のMFPを制御するコントローラ100を図29に示す。コントローラ100は、システムバスブリッジ(SBB)101により、CPU102、メモリコントローラ(MC)103、汎用バス105、画像処理部110、画像データ展開部(RIP(Raster Image Processor))113が接続されている。

【0005】

汎用バス105には、画像データを蓄積するための：大容量記憶部(HDD(ハードディスクドライブ))107を制御するハードディスクコントローラ(HDDCont)106、MFPが接続されているネットワーク108を介して、外部機器間との画像データの転送を行うインターフェースになるネットワークI/F109が接続されている。この

50

画像データとしては、ページベクタ形式（PDL（ページ記述言語）、PDF、SVG等）の画像データがある。

【0006】

HDDCont106には、HDD（ハードディスクドライブ）107が接続されており、画像データの記憶媒体として使われる。同様に、MC103には、システムメモリ（Memory）104が接続されており、画像データを一時記憶するための媒体として使われる。システムメモリ104には、一般的には、DIMMが用いられる。

【0007】

画像処理部110には、スキャナ111及びプリンタ112が接続されている。スキャナ111から入力された画像データは、画像処理部110によって所定の画像処理が実施された後、コントローラ100へ入力される。また、コントローラ100内に記憶されている画像データは、画像処理部110によって所定の画像処理が実施され、プリンタ112へ出力される。

【0008】

コントローラ100でハンドリングされる画像データは、ネットワーク経由での外部機器との入出力はページベクタ形式（PDLやPDF、SVG等）、スキャナ111やプリンタ112との入出力はラスタデータ形式でインターフェースされる。外部機器から入力されたページベクタ形式の画像データは、CPU102により、プリミティブなオブジェクトにインタプリットされ、DL（Display List）と呼ばれる中間データ（DLデータ）に変換されてから、RIP113に入力される。

【0009】

これらの画像データはコントローラ100の内部のシステムメモリ104に一旦記憶される。そのため、システムメモリ104上には、ラスタデータ、ページベクタデータ（PDL等）、DLデータ等の多種類のデータが存在する。

【0010】

HDD107には、画像データとしてスキャナ113から入力された画像データとRIP113でレンダリングされたラスタ画像データが記憶される。

【0011】

【特許文献1】特開2004-120639号公報

【発明の開示】

【発明が解決しようとする課題】

【0012】

上述のようなMFPが扱う画像データの中でも、ラスタ画像データはデータサイズが大きい。そのため、システムメモリ104のメモリサイズや、汎用バス105及びHDDCont106-HDD107間のバンド幅等のシステム・リソースを多く消費する。

【0013】

これに加え、PDLデータ等のページベクタデータでは、システム中でインタプリットを行い、描画オブジェクトを生成するDLデータに展開する。その際、DLデータをシステムメモリ104にスプールするため、それによるメモリリソースの消費量も莫大なものとなっている。

【0014】

一方、最近ではユーザの出力画像の画質要求がますます高くなり、その解決策の1つとして、画像データに対する高解像度化（高画質化）が促進されている。また、画質と並行してシステムの処理速度向上も要求されている。

【0015】

そのため、上述の様々な要求仕様を満足するために必要なシステム・リソースが肥大化してしまう。従って、プロダクトのコストパフォーマンスでの折り合いをつけることが課題となっている。

【0016】

また、このプロダクトのコストパフォーマンスに対する課題とは別に、複雑、多様化す

10

20

30

40

50

るシステムを開発するための人的リソースの問題も解決することが要求されている。そして、これを満足するためには、様々な要求仕様に対し、1つの基本システムをスケーラブルな形で構成することにより、製品ラインナップを効率良く整備することが課題となっている。

【0017】

例えば、図29における画像処理部110やRIP113等のモジュールを、ハイエンドな機種では複数個実装し、分散処理できるようなシステムが必要となってくる。

【0018】

他方、オフィスにおいてはペーパーレス化が進行しており、紙の出力物と電子データをシームレスに扱えるような要求も生まれてきている。そのためには、紙と電子データとのI/F機器であるMFPにおいても、例えば、蓄積画像データの検索性の向上や、ラスト画像データをオブジェクト化して再利用可能なオブジェクトデータに変換する、POD(Print On Demand)印刷に対応するために画像加工を高速化する、といったよりインテリジェントな機能を持つことが必要となっている。

【0019】

本発明は上記の課題を解決するためになされたものであり、システム全体のシステム・リソースの制約を緩和し、かつトータルスループットを向上することができる画像処理装置及びその制御方法、プログラムを提供することを目的とする。

【課題を解決するための手段】

【0020】

上記課題を解決するための本発明による画像処理装置は以下の構成を備える。即ち、入力された画像データに対する処理を実行する画像処理装置であって、
画像データを入力する入力手段と、
画像データを出力する出力手段と、
ラスト画像データを、所定の大きさのブロック毎に分割し、ベクトル化処理を実行することにより、当該所定の大きさに分割された各ブロックに対応するブロックベクタ画像データに変換する第1変換手段と、

ブロックベクタ画像データを記憶する記憶手段と、

ブロックベクタ画像データをラスト画像データに展開する展開手段と、

前記入力手段から入力されたラスト画像データを前記第1変換手段でブロックベクタ画像データに変換させ、当該変換されたブロックベクタ画像データを前記記憶手段に記憶させ、前記記憶手段に記憶されているブロックベクタ画像データを前記展開手段で展開して得られるラスト画像データを前記出力手段から出力させるように、装置内での処理対象の画像データの転送を制御する画像データ転送制御手段と、

1ページ分のブロックベクタ画像データから、ページ全体を示すページベクタ画像データへ変換する第3変換手段とを備え、

前記展開手段は、複数のブロックベクタ画像データを並列に処理するための、複数の小展開部を有し、

前記画像データ転送制御手段は、

前記出力手段から出力させるべき画像データの形式がラスト画像データ形式であると判断した場合は、前記記憶手段に記憶されたブロックベクタ画像データに対して前記展開手段を実行させて得られるラスト画像データを前記出力手段から出力させ、

前記出力手段から出力させるべき画像データの形式がベクタ画像データ形式であると判断した場合は、前記記憶手段に記憶されたブロックベクタ画像データに対して前記第3変換手段を実行させて得られる前記ページベクタ画像データを前記出力手段から出力させる

ように、装置内での処理対象の画像データの転送を制御する。

【0021】

また、好ましくは、入力された画像データに対する処理を実行する画像処理装置であって、

画像データを入力する入力手段と、
画像データを出力する出力手段と、
ラスト画像データを、所定の大きさのブロック毎に分割し、ベクトル化処理を実行することにより、当該所定の大きさに分割された各ブロックに対応するブロックベクタ画像データに変換する第1変換手段と、
ブロックベクタ画像データを記憶する記憶手段と、
ブロックベクタ画像データをラスト画像データに展開する展開手段と、
前記入力手段から入力されたラスト画像データを前記第1変換手段でブロックベクタ画像データに変換させ、当該変換されたブロックベクタ画像データを前記記憶手段に記憶させ、前記記憶手段に記憶されているブロックベクタ画像データを前記展開手段で展開して得られるラスト画像データを前記出力手段から出力させるように、装置内での処理対象の画像データの転送を制御する画像データ転送制御手段とを備え、
前記展開手段は、複数のブロックベクタ画像データを並列に処理するための、複数の小展開部と、前記複数の小展開部に対して共通のフォントキャッシュ部とを備え、
前記フォントキャッシュ部は、フォントキャッシュメモリと、前記複数の小展開部それぞれが前記フォントキャッシュメモリを参照中であることを示すロックフラグを記憶するロックフラグレジスタを備える。

10

【0039】

上記課題を解決するための本発明による画像処理装置の制御方法は以下の構成を備える。即ち、

20

入力された画像データに対する処理を実行する画像処理装置の制御方法であって、
ラスト画像データを、所定の大きさのブロック毎の分割とベクトル化処理とにより、所定の大きさに分割されたブロックベクタ画像データに変換する第1変換工程と、

ブロックベクタ画像データを記憶手段に記憶する記憶工程と、

ブロックベクタ画像データをラスト画像データに展開する展開工程と、

前記入力されたラスト画像データを前記第1変換工程でブロックベクタ画像データに変換させ、当該変換されたブロックベクタ画像データを前記記憶手段に記憶させ、前記記憶手段に記憶されているブロックベクタ画像データを前記展開工程で展開して得られるラスト画像データを出力部から出力させるように、装置内での処理対象の画像データの転送を制御する画像データ転送制御工程と、

30

1ページ分のブロックベクタ画像データから、ページ全体を示すページベクタ画像データへ変換する第3変換工程とを備え、

前記展開工程は、複数の小展開部によって、複数のブロックベクタ画像データを並列に処理し、

前記画像データ転送制御工程は、

前記出力部から出力させるべき画像データの形式がラスト画像データ形式であると判断した場合は、前記記憶手段に記憶されたブロックベクタ画像データに対して前記展開工程を実行させて得られるラスト画像データを前記出力部から出力させ、

前記出力部から出力させるべき画像データの形式がベクタ画像データ形式であると判断した場合は、前記記憶手段に記憶されたブロックベクタ画像データに対して前記第3変換工程を実行させて得られる前記ページベクタ画像データを前記出力部から出力させるように、装置内での処理対象の画像データの転送を制御する。

40

【0040】

上記課題を解決するための本発明によるプログラムは以下の構成を備える。即ち、
入力された画像データに対する処理をコンピュータに実行させる画像処理装置の制御を実現するプログラムであって、

ラスト画像データを、所定の大きさのブロック毎の分割とベクトル化処理とにより、所定の大きさに分割されたブロックベクタ画像データに変換する第1変換工程と、

ブロックベクタ画像データを記憶手段に記憶する記憶工程と、

ブロックベクタ画像データをラスト画像データに展開する展開工程と、

50

前記入力されたラスタ画像データを前記第 1 変換工程でブロックベクタ画像データに変換させ、当該変換されたブロックベクタ画像データを前記記憶手段に記憶させ、前記記憶手段に記憶されているブロックベクタ画像データを前記展開工程で展開して得られるラスタ画像データを出力部から出力させるように制御する制御工程と、

1 ページ分のブロックベクタ画像データから、ページ全体を示すページベクタ画像データへ変換する第 3 変換工程とを備え、

前記展開工程は、複数の小展開部によって、複数のブロックベクタ画像データを並列に処理し、

前記制御工程は、

前記出力部から出力させるべき画像データの形式がラスタ画像データ形式であると判断した場合は、前記記憶手段に記憶されたブロックベクタ画像データに対して前記展開工程を実行させて得られるラスタ画像データを前記出力部から出力させ、

前記出力部から出力させるべき画像データの形式がベクタ画像データ形式であると判断した場合は、前記記憶手段に記憶されたブロックベクタ画像データに対して前記第 3 変換工程を実行させて得られる前記ページベクタ画像データを前記出力部から出力させる

ように、装置内での処理対象の画像データの転送を制御する。

【発明の効果】

【0041】

本発明によれば、システム全体のシステム・リソースの制約を緩和し、かつトータルスループットを向上することができる画像処理装置及びその制御方法、プログラムを提供できる。

【発明を実施するための最良の形態】

【0042】

以下、本発明の実施の形態について図面を用いて詳細に説明する。

【0043】

[実施形態 1]

[MFP 装置概要]

図 1 は本発明の実施形態 1 の画像処理システムを構成する MFP のコントローラの詳細を示すブロック図である。

【0044】

MFP 1000 を制御するコントローラ 1 は、システムバスブリッジ (SBB) 2 により、CPU 3、メモリコントローラ (MC) 4、汎用バス 6、タイル/ページベクタ変換部 13、ラスタ/ベクタ変換部 14、画像処理部 15、画像データ展開部 (RIP) 18 が接続されている。

【0045】

ここで、この RIP 18 は、タイルベクタデータを展開することが可能であり、内部には、複数の小画像データ展開部 (μ RIP) 18a ~ 18d が構成されている。

【0046】

MC 4 には、システムメモリ (Memory) 5 が接続されており、画像データを一時記憶するための媒体として使われる。

【0047】

汎用バス 6 には、画像データを蓄積するための HDD 8 を制御するハードディスクコントローラ (HDDC) 7 及び操作部 (例えば、LCD 等から構成されるタッチパネル) 10 を制御する操作部コントローラ 9 及び、MFP 1000 が接続されているネットワーク 12 を介して、外部機器間との画像データの転送を行うインターフェースになるネットワーク I/F 11 が接続されている。

【0048】

尚、操作部 10 では、実施形態 1 及び後述する各実施形態の各種処理の実行指示の入力や、処理結果等を表示するための操作画面が表示され、ユーザは、この操作画面を介して各種操作を実現することができる。

10

20

30

40

50

【 0 0 4 9 】

ラスト/タイルベクタ変換部 1 4 には、画像処理部 1 5 が接続されている。また、画像処理部 1 5 には、スキャナ 1 6 及びプリンタ 1 7 が接続されている。

【 0 0 5 0 】

また、S B B 2 には、R I P 1 8 が接続されている。また、R I P 1 8 には、R I P 1 8 から出力されるデータを記憶するローカルメモリ (L o c a l M e m o r y) 1 9 が接続されている。

【 0 0 5 1 】

コントローラ 1 でハンドリングされる画像データは、外部機器との入出力はベクタ (P D L、P D F、S V G など) 形式の画像データ (以下、ベクタデータとも言う)、スキャナ 1 6 やプリンタ 1 7 との入出力はラスト形式の画像データ (以下、ラストデータとも言う) でインターフェースされる。

10

【 0 0 5 2 】

コントローラ 1 では、スキャンデータ (ラストデータ) はラスト/タイルベクタ変換部 1 4 でタイルベクタデータに変換される。また、このタイルベクタデータから R I P 1 8 の処理によって得られるタイル D L データは、R I P 1 8 に接続されたローカルメモリ 1 9 に記憶される。

【 0 0 5 3 】

従って、システムメモリ 5 上には、ページベクタデータとタイルベクタデータの 2 種類の画像のみが記憶される。つまり、画像サイズの大きいラストデータ及び D L データをシステムメモリ 5 に記憶する必要がなくなる。そのため、システムメモリ 5 上で確保しなければならない画像データ領域を削減することができる。

20

【 0 0 5 4 】

また、R I P 1 8 から出力される D L データはタイル単位に分割された、タイル D L データで記憶される。そのため、従来のページ単位のページ D L データに比べ、非常に少ないメモリ容量で記憶できる。従って、ローカルメモリ 1 9 はオンチップ上に実装することが可能になり、メモリレイテンシを小さくできる。その結果、タイルデータ展開速度を高速化することが可能となる。

【 0 0 5 5 】

また、タイルベクタデータのみを画像データとして、H D D 8 上に記憶すれば良いので、H D D 8 へのアクセス速度のボトルネックが緩和され、データ処理の高速化が図れる。同時に、タイル単位で処理することにより、R I P 1 8 のコストダウンも可能となる。

30

【 0 0 5 6 】

より高い処理能力が要求される場合は、R I P 1 8 内に備える μ R I P 1 8 a ~ 1 8 d を並列に複数実装することで、処理能力を可変にすることができる。このようにすることで、コントローラ 1 の処理能力がシンプルに調整できることから、スケーラビリティの確保が容易なシステムを構築できる。

【 0 0 5 7 】

尚、本発明では、ネットワーク I / F 1 1 及びスキャナ 1 6 がコントローラ 1 内へ画像データを入力する画像入力部として機能する。また、ネットワーク I / F 1 1 及びプリンタ 1 7 が画像データを出力する画像出力部として機能する。

40

【 0 0 5 8 】

以下に、M F P 1 0 0 0 が実現可能な各種処理のデータフローについて説明する。

【 0 0 5 9 】

[コピー]

図 2 は本発明の実施形態 1 の画像処理システムでのコピー動作に係るデータフローを示す図である。

【 0 0 6 0 】

尚、このデータフローは、C P U 3 の制御の下、M F P 1 0 0 0 を構成する各種構成要素を協調動作させることで実現される。

50

【 0 0 6 1 】

また、図 2 中に示す矢印は、各種データフローを示している。特に、実線矢印はラスタデータ（ラスタ画像データ）、破線矢印はタイルベクタデータ（タイルベクタ画像データ）、1 点鎖線矢印はページベクタデータ（ページベクタ画像データ）のデータフローを示している。ページベクタデータ及びタイルベクタデータについては、後述のタイル／ページベクタ変換部 1 3 で詳細に説明する。

【 0 0 6 2 】

（ S 2 1 ）：操作部 1 0 より、ユーザがコピー開始を指示すると、スキャナ 1 6 は原稿画像の読取動作を開始する。スキャナ 1 4 より画像処理部 1 5 へ入力された画像（ R、G、B 画像）は画像処理部 1 5 のクロック同期に周波数変換された後、例えば、以下の処理が実行される。

10

【 0 0 6 3 】

1) スキャナ 1 6 内の C C D センサのラインピッチや色収差等のスキャナ特性の補正処理

2) 色空間補正やシャープネス等の入力画像データの画質補正処理

3) 入力画像データの枠消やブック枠消等の画像加工処理

（ S 2 2 ）：画像処理部 1 5 による画像処理が終了し、画像処理部 1 5 から出力された画像データはラスタ／タイルベクタ変換部 1 4 へ入力され、タイルベクタ変換処理が実行される。即ち、ラスタ／タイルベクタ変換部 1 4 は、画像データを所定の大きさのブロック（タイル）に分割する。そして、各ブロック内のラスタデータに対して、ベクトル化処理を実行して、ブロック（タイル）単位のベクタデータ（タイルベクタデータ（ブロックベクタデータ））を生成する。

20

【 0 0 6 4 】

生成されたタイルベクタデータは、S B B 2 によりバスの調停を受け、システムメモリ 5 へのバス権を取得し、M C 4 を介して、システムメモリ 5 に記憶される。（尚、S B B 2 経由でデータバスが接続される場合は、基本的にバスの調停を受け、バス権を取得する手続きを踏むが、以降のフロー説明では省略する。）

（ S 2 3 ）：システムメモリ 5 に記憶されたタイルベクタデータは、H D D C o n t 7 と M C 4 を介して、S B B 2 経由で H D D 8 に記憶される。H D D 8 にタイルベクタデータを記憶することにより、複数部の原稿をコピーする時にソーティングをして、ページ順を変えて出力したり、M F P 1 0 0 0 内に保存画像データとして記憶したりすることができる。

30

【 0 0 6 5 】

（ S 2 4 ）：H D D 8 に記憶されたタイルベクタデータは、プリンタ 1 7 内のプリンタ C P U （不図示）から送られてくるプリンタレディのタイミングに合わせて、H D D C o n t 7 により読み出され、S B B 2、M C 4 を経由してシステムメモリ 5 に一時的に記憶される。

【 0 0 6 6 】

仮に、読み取ったタイルベクタデータを H D D 8 からダイレクトにプリンタ 1 7 へ出力する場合、H D D 8 のアクセススピードが律則したり、汎用バス 6 のバスの混雑度合によりプリンタ 1 7 に同期して出力することが保証できなくなる。そのため、プリンタ 1 7 に同期してデータ転送を行う前に、システムメモリ 5 にページベクタデータをスプールしておくことで、リアルタイムなスループットを保証する。

40

【 0 0 6 7 】

（ S 2 5 ）：システムメモリ 5 に記憶されたタイルベクタデータは、プリンタ 1 7 からコントローラ 1 に送られる起動信号に従って、M C 4 によって読み出され、S B B 2 を介して R I P 1 8 に転送される。

【 0 0 6 8 】

R I P 1 8 では、まず、タイルベクタデータを解析し、タイル単位の描画オブジェクト（タイル D L データ）の生成（インタプリット）を実行する。生成されたタイル D L デー

50

タはローカルメモリ 19 に一旦記憶される。

【0069】

RIP18 はローカルメモリ 19 からタイルDLデータを読み出して、タイル単位のラスタデータ（タイルラスタデータ）へと展開し、出力する。

【0070】

実施形態 1 では、上述のように、RIP18 内に 4 つの小画像データ展開部（ μ RIP）18a ~ 18d を備えている。コントローラ 1 は、 μ RIP18a ~ μ RIP18d を並列に動作させることにより、タイルベクタデータの展開を高速に行わせることができる。

【0071】

ここで、画像処理システムの全体パフォーマンスは、ベクタデータ展開時間が支配的であり、この μ RIP の構成数を増やすことで、パフォーマンスアップが見込める。そのため、本発明のような構成を用いると、その構成数あるいは動作させる構成数を増減させることで、容易にスケラブルなシステムを構築することが可能となる。

【0072】

（S26）：RIP18 によって生成されたタイルラスタデータは、画像処理部 15 に転送され、例えば、以下の処理が実行される。

【0073】

- 1) タイルラスタデータからページラスタデータへの変換処理
 - 2) プリンタの特性に合わせた出力画像の色や濃度の補正処理
 - 3) 画像データを量子化して出力画像の階調変換を行う中間調処理
 - 4) プリンタ I/F クロックに同期して画像を出力するための周波数変換処理
- そして、画像処理部 15 で、1) ~ 4) の画像処理が実行されて得られたラスタデータは、プリンタ 17 に転送され、記録媒体上に印刷され出力される。

【0074】

[プリント]

図 3 は本発明の実施形態 1 の画像処理システムでのプリント動作に係るデータフローを示す図である。

【0075】

尚、このデータフローは、CPU3 の制御の下、MFP1000 を構成する各種構成要素を協調動作させることで実現される。

【0076】

（S31）：ネットワーク 12 に接続された外部機器より、汎用バス 6 に接続されたネットワーク I/F 11 がページベクタデータを受信する。そして、SBB2 の先に接続された MC4 を介してシステムメモリ 5 に転送する。

【0077】

（S32）：システムメモリ 5 に記憶されたページベクタデータは、タイル/ページベクタ変換部 13 より読み出され、タイルベクタ変換処理が実行される。即ち、タイル/ページベクタ変換部は、ページベクタデータ内に存在するオブジェクトを所定の大きさのブロック（タイル）内に収まるオブジェクトに分割する。そして、タイル単位のベクタデータ（タイルベクタデータ）を生成する。

【0078】

（S33）：生成されたタイルベクタデータは、SBB2 を介して再度システムメモリ 5 に記憶される。

【0079】

（S34）：システムメモリ 5 に記憶されたタイルベクタデータは、HDDCont7 と MC4 を介して、SBB2 経由で HDD8 に記憶される。HDD8 にタイルベクタデータを記憶することにより、複数部の原稿をコピーする時にソーティングをして、ページ順を変えて出力したり、MFP1000 内に保存画像データとして記憶したりすることができる。

10

20

30

40

50

【 0 0 8 0 】

(S 3 5) : H D D 8 に記憶されたタイルベクタデータは、プリンタ 1 7 内の C P U (不図示) から送られてくるプリンタレディのタイミングに合わせて、H D D C o n t 7 により読み出され、S B B 2、M C 4 を経由してシステムメモリ 5 に一時的に記憶される。

【 0 0 8 1 】

読み取ったタイルベクタデータを H D D 8 からダイレクトにプリンタ 1 7 へ出力する場合、H D D 8 のアクセススピードが律則したり、汎用バス 6 のバスの混雑度合によりプリンタ 1 7 に同期して出力することが保証できなくなる。そのため、プリンタ 1 7 に同期してデータ転送を行う前に、システムメモリ 5 に 1 ページ分のベクタ画像データをスプールすることにより、リアルタイムなスループットを保証する。

10

【 0 0 8 2 】

(S 3 6) : システムメモリ 5 に記憶されたタイルベクタデータは、プリンタ 1 7 からコントローラ 1 に送られる起動信号に従って、M C 4 によって読み出され、S B B 2 を介して R I P 1 8 に転送される。

【 0 0 8 3 】

R I P 1 8 では、まず、タイルベクタデータを解析し、タイル単位の描画オブジェクト (タイル D L データ) の生成 (インタプリット) を実行する。生成されたタイル D L データはローカルメモリ 1 9 に一旦記憶される。

【 0 0 8 4 】

R I P 1 8 はローカルメモリ 1 9 からタイル D L データを読み出して、タイル単位のラスタデータ (タイルラスタデータ) へと展開し、出力する。

20

【 0 0 8 5 】

(S 3 7) : R I P 1 8 によって生成されたタイルラスタデータは、画像処理部 1 5 に転送され、例えば、以下の処理が実行される。

【 0 0 8 6 】

- 1) タイルラスタデータからページラスタデータへの変換処理
- 2) プリンタの特性に合わせた出力画像の色や濃度の補正処理
- 3) 画像データを量子化して出力画像の階調変換を実行する中間調処理
- 4) プリンタ E / F クロックに同期して画像を出力するための周波数変換処理

そして、画像処理部 1 5 で、1) ~ 4) の画像処理が実行されて得られたラスタデータは、プリンタ 1 7 に転送され、記録媒体上に印刷され出力される。

30

【 0 0 8 7 】

[送信]

図 4 は本発明の実施形態 1 の画像処理システムでの送信動作に係るデータフローを示す図である。

【 0 0 8 8 】

尚、このデータフローは、C P U 3 の制御の下、M F P 1 0 0 0 を構成する各種構成要素を協調動作させることで実現される。

【 0 0 8 9 】

また、画像データを H D D 8 に格納するまでのデータフローについては、ラスタデータの場合は [コピー] と、ネットワーク 1 2 上の外部機器からの入力されたページベクタデータの場合は [プリント] と同一であるので、その説明は割愛する。

40

【 0 0 9 0 】

尚、画像データを H D D 8 に格納する処理は、ユーザからの格納指示によって実行されるようにしてもよいし、[コピー] や [プリント] の処理の際に H D D 8 に自動的に残しておくようにしてもよい。このようにして、H D D 8 に格納された画像データの中から、ユーザによって指定された画像データを送信するように指示された際に行われる送信処理について説明する。

【 0 0 9 1 】

(S 4 1) : H D D 8 に記憶されたタイルベクタデータは S B B 2 を介して、汎用バス

50

6に接続されたHDD Cont 7より読み出され、システムメモリ5に一時的に記憶される。

【0092】

(S42): システムメモリ5に記憶されたタイルベクタデータは、タイル/ページベクタ変換部13より読み出され、タイルベクタ変換処理を実行する。即ち、ブロック単位に分割されたオブジェクトを結合し、ページ全体でオブジェクトを記述したページベクタデータを生成する。つまり、1ページ分のタイルベクタデータを、ページ全体のベクタデータを示すページベクタデータを生成する。

【0093】

(S43): 生成されたページベクタデータは、SBB2を介して再度システムメモリ5に記憶される。

10

【0094】

(S44): システムメモリ5中に記憶されたページベクタデータは、汎用バス6に接続されたネットワークI/F11から読み出され、ネットワーク12に接続された外部機器へと送信転送される。

【0095】

本発明のように、外部機器に送信する際に、タイルベクタデータをページベクタデータに戻して、そのデータを構成するオブジェクト数を減らすことで、送信データ量を削減することができる。また、PDFやSVG等の汎用フォーマットへ容易に変換することができる。

20

【0096】

尚、本発明では、スキャナ16から入力したラスタデータを、外部機器に送信することも可能である。この場合は、そのラスタデータをページベクタに変換してから、外部機器へ送信することが好ましい。

【0097】

[ラスタ/タイルベクタ変換部]

次に、ラスタ/タイルベクタ変換部14の処理の詳細について説明する。

【0098】

図5は本発明の実施形態1のラスタ/タイルベクタ変換部が実行する処理を示すフローチャートである。

30

【0099】

(ステップS51: ブロックセレクション(領域分割: BS)処理)

画像処理部15より入力されたラスタデータ(イメージデータ)を、文字あるいは線画を含む文字・線画領域と、ハーフトーンの写真領域、不定形の画像領域その他に分割する。さらに、文字・線画領域については、主に文字を含む文字領域と、主に表、図形等を含む線画領域とに分離し、線画領域は表領域と図形領域に分割する。

【0100】

尚、実施形態1では、処理対象中の画像の連結画素を検知し、その連結画素の外接矩形領域の形状・サイズ・画素密度等の特徴量を用いて、属性毎の領域に分離するものとするが、その他の領域分割手法を用いても構わない。

40

【0101】

文字領域については、文字段落ごとの纏まった塊をブロックとして矩形ブロック(文字領域矩形ブロック)にセグメント化する。線画領域では、表、図形等の個々のオブジェクト(表領域矩形ブロック、線画領域矩形ブロック)ごとに矩形ブロックにセグメント化する。

【0102】

ハーフトーンで表現される写真領域は、画像領域矩形ブロック、背景領域矩形ブロック等のオブジェクトごとに、矩形ブロックにセグメント化する。

【0103】

分離された各領域は、さらに所定の大きさの領域(タイル)単位に分割され、タイル単

50

位で、次のベクトル化処理でベクトル化される。

【 0 1 0 4 】

(ステップ S 5 2 : ベクトル化処理)

ベクトル化処理により、各属性の領域のイメージデータをベクトルデータに変換する(ベクトル化する)。ベクトル化の方法は、例えば、以下、方法(a)~(f)等が存在する。

【 0 1 0 5 】

(a) 属性領域が文字領域のとき、さらにOCRによる文字画像のコード変換を行ったり、あるいは文字のサイズ、スタイル、字体を認識し、原稿を走査して得られた文字に可視的に忠実なフォントデータに変換する。

10

【 0 1 0 6 】

(b) 属性領域が文字領域であり、かつOCRによる認識が不可能であったとき、文字の輪郭を追跡し、輪郭情報(アウトライン)を線分のつながりとして表現する形式に変換する。

【 0 1 0 7 】

尚、実施形態1では、文字領域に対して、OCR結果に応じて、方法(a)と方法(b)とのいずれかを用いる例を示したが、これに限るものではなく、例えば、方法(a)は用いずに全ての文字領域に対して方法(b)だけを用いるようにしても構わない。

【 0 1 0 8 】

(c) 属性領域が図形領域のとき、図形オブジェクトの輪郭を追跡し、輪郭情報を線分のつながりとして表現する形式に変換する。

20

【 0 1 0 9 】

(d) 方法(b)及び(c)の線分形式のアウトライン情報をベジエ関数等でフィッティングして、関数情報に変換する。

【 0 1 1 0 】

(e) 方法(c)の図形オブジェクトの輪郭情報から、図形の形状を認識し、円、矩形、多角形、等の図形定義情報に変換する。

【 0 1 1 1 】

(f) 属性領域が図形領域であって、特定領域の表形式のオブジェクトの場合、罫線や枠線を認識し、所定のフォーマットの帳票フォーマット情報に変換する。

30

【 0 1 1 2 】

(ステップ S 5 3 : タイルベクタデータ生成処理)

ステップS52で、方法(a)~(f)のフォーマットコード情報、図形情報、関数情報等のコマンド定義形式情報にベクトル変換されたデータに対し、コントローラ1内でページベクタデータかタイルベクタデータかを判別するベクタタイプや、当該タイルのページ内の座標位置等の座標情報を判別するためのヘッダ情報を付加した、タイルベクタデータを生成する。このようにして、タイル単位に各種情報が付加されたタイルベクタデータをSBB2へ出力する。

【 0 1 1 3 】

(ステップ S 5 4 : 終了判定処理)

40

処理対象のラスタデータの有無を判定する。処理対象のラスタデータがある場合(ステップS54でNO)、ステップS51に戻る。一方、処理対象のラスタデータがない場合(ステップS54でYES)、処理を終了する。

【 0 1 1 4 】

[タイル / ページベクタ変換部]

次に、タイル / ページベクタ変換部13の処理の詳細を説明するにあたり、処理対象となるドキュメントデータ(画像データ)について説明する。

【 0 1 1 5 】

図6は本発明の実施形態1のネットワークから転送されてくるドキュメントデータの一例を示している。

50

【 0 1 1 6 】

図 6 では、ドキュメントデータ 8 0 1 の短手方向を「 X 」方向、長手方向を「 Y 」方向とするデバイス座標系を定義する。ドキュメントデータ 8 0 1 としては、ページベクタデータ、タイルベクタデータ、あるいはラスタデータ表現を含むページベクタデータ（タイルベクタデータ）、あるいはラスタデータのいずれかで構成される可能性がある。

【 0 1 1 7 】

ここで、ドキュメントデータ 8 0 1 が、ページベクタデータである場合、その内容を構成する記述例について、図 7 を用いて説明する。

【 0 1 1 8 】

図 7 は本発明の実施形態 1 のページベクタデータの記述例を示す図である。

10

【 0 1 1 9 】

図 7 において、9 0 1 はドキュメントデータ全体の設定に関わるドキュメント設定命令部分、9 0 2 は文字の描画命令部分、9 0 3 は図形の描画命令部分を示している。

【 0 1 2 0 】

各描画命令部分の詳細について説明する。

【 0 1 2 1 】

ドキュメント設定命令部分 9 0 1 において、C 1 ~ C 5 は、ドキュメント全体に関するコマンドである。従って、これらのコマンド C 1 ~ C 5 は、ドキュメント 1 部について 1 ヶ所しか付いていない。

【 0 1 2 2 】

20

これらドキュメントデータ全体に関係するコマンドには、例えば、キャラクタセットコマンド（フォント指定コマンド）、スケーラブルフォントコマンド（スケーラブルフォントを使用するか否かを指定するコマンド）、ハードリセットコマンド（以前のプリンタ使用環境をリセットするコマンド）等がある。

【 0 1 2 3 】

ここで、C 1 はドキュメント設定開始コマンドである。C 2 はドキュメントデータの出力用紙サイズを示すコマンドであり、この場合には、A 4 の設定になっている。C 3 はドキュメントデータの方向を示すコマンドである。ここでは、ポートレートとランドスケープがあるが、この場合にはポートレート（P O R T）の設定になっている。

【 0 1 2 4 】

30

C 4 はドキュメントデータのタイプを示すコマンドであり、ページベクタで構成されるドキュメントデータであるかタイルベクタで構成されるドキュメントデータであるかを示している。この場合には、ページ（P A G E）に設定されている。C 5 はドキュメント設定終了コマンドである。

【 0 1 2 5 】

文字の描画命令部分 9 0 2 及び図形の描画命令部分 9 0 3 を構成する C 6 ~ C 2 2 は、ドキュメントデータを出力するための各種コマンドである。

【 0 1 2 6 】

C 6 はページの開始を示すコマンドである。C 7 は文字のフォントの種類を選択するためのコマンドであり、この場合には「 1 」という番号の付けられたフォントセットに設定されている。C 8 はフォントの大きさを設定するコマンドであり、この場合には「 1 0 ポイント」の大きさに設定されている。

40

【 0 1 2 7 】

C 9 は文字の色を設定するコマンドであり、順に R（赤）、G（緑）、B（青）各色成分の輝度を示してある。この輝度は、例えば、0 から 2 5 5 の 2 5 6 段階で量子化されているものとする。この場合は、{ 0 , 0 , 0 } に設定されている。C 1 0 は文字を描画する開始位置の座標を示すコマンドである。座標位置（X , Y）は、ページの左上を原点に指定する。この場合は、ページの { 1 0 , 5 } の位置から文字の描画を開始するように設定されている。C 1 1 は実際に描画する文字列（X X X X . . . Y Y . . .）を示すコマンドである。

50

【 0 1 2 8 】

C 1 2 は図形描画の際の面の塗りつぶしの色を示すコマンドである。色の指定は、文字の色と同様である。C 1 3 は図形描画の線の色を指定するコマンドである。C 1 4 は図形描画位置の座標を示すコマンドである。

【 0 1 2 9 】

C 1 5 は円弧を描画する際の半径を指定するコマンドであり、この場合、「 1 0 」座標単位を示している。C 1 6 は閉円弧の描画をするコマンドである。コマンド内の 2 個のパラメータは円弧を描画する際の描画開始角度と終了角度を示している。垂直情報を 0 度として、この場合には 0 度から 9 0 度の円弧を描画することを示している。

【 0 1 3 0 】

C 1 7 ~ C 2 1 は、C 1 2 ~ C 1 6 までのコマンドと同様に、図形描画の際の面、線の色指定、位置の指定等のコマンドである。C 2 2 はページの終了を示すコマンドである。

【 0 1 3 1 】

一方、ドキュメントデータ 8 0 1 がタイルベクタデータである場合について、図 8 を用いて説明する。

【 0 1 3 2 】

図 8 は本発明の実施形態 1 のタイルベクタデータの例を示す図である。

【 0 1 3 3 】

図 8 では、図 6 のドキュメントデータ 8 0 1 (ページベクタデータ) を、ブロック (タイル) 単位で分割したタイルベクタデータ (ドキュメントデータ 1 0 0 1) の一例を示している。

【 0 1 3 4 】

図 8 では、ドキュメントデータ 1 0 0 1 の短手方向を「 X 」方向、長手方向を「 Y 」とするデバイス座標系を定義する。また、図中の X 方向に配列された数列は、X 方向のタイル ID、Y 方向に配列された数列は、Y 方向のタイル ID を表す。A ~ D は、それぞれタイル ID = (0 , 0)、(1 , 0)、(2 , 4)、(1 , 5) の位置にあるタイルデータを示している。

【 0 1 3 5 】

ここで、このドキュメントデータ 1 0 0 1 が、タイルベクタデータである場合、その内容を構成する記述例について、図 9 を用いて説明する。

【 0 1 3 6 】

図 9 は本発明の実施形態 1 のタイルベクタデータの記述例を示す図である。

【 0 1 3 7 】

図 9 において、1 1 0 1 はドキュメントデータ全体の設定に関わるドキュメント設定命令部分、1 1 0 2 は描画命令部分全体、1 1 0 3 ~ 1 1 0 6 はそれぞれタイル A , B , C , D の描画命令部分を示している。1 1 0 7、1 1 0 8 はそれぞれタイル D の文字の描画命令部分、図形の描画命令部分を示している。

【 0 1 3 8 】

各描画命令の詳細について説明する。

【 0 1 3 9 】

ドキュメント設定命令部分 1 1 0 1 において、C 1 ~ C 5 は、ドキュメント全体に関係するコマンドである。従って、これらのコマンド C 1 ~ C 5 は、ドキュメント 1 部について 1 ケ所しか付いていない。

【 0 1 4 0 】

これらドキュメントデータ全体に関係するコマンドには、例えば、キャラクタセットコマンド (フォント指定コマンド)、スケーラブルフォントコマンド (スケーラブルフォントを使用するか否かを指定するコマンド)、ハードリセットコマンド (以前のプリンタ使用環境をリセットするコマンド) 等がある。

【 0 1 4 1 】

10

20

30

40

50

ここで、C 1 はドキュメント設定開始コマンドである。C 2 はドキュメントデータの出力用紙サイズを示すコマンドであり、この場合には、A 4 の設定になっている。C 3 はドキュメントデータの方向を示すコマンドである。ここでは、ポートレートとランドスケープがあるが、この場合にはポートレート (P O R T) の設定になっている。

【 0 1 4 2 】

C 4 はドキュメントデータのタイプを示すコマンドであり、ページベクタで構成されるドキュメントデータあるのかタイルベクタで構成されるドキュメントデータあるかを示している。この場合には、タイル (T I L E) に設定されている。C 5 はドキュメント設定終了コマンドである。

【 0 1 4 3 】

描画命令部分 1 1 0 2 を構成する C 6 ~ C 5 0 0 は、ドキュメントデータを出力するための各種コマンドである。

【 0 1 4 4 】

C 6 はページの開始を示すコマンドである。C 7 は図 8 のタイル A の描画コマンドの開始を示すコマンドである。ここで、T i l e S t a r t (0 , 0) 中の 2 個のパラメータは、ドキュメントデータ内におけるタイル I D を示している。C 8 はタイル A の描画コマンドの終了を示すコマンドである。タイル A のように、そのタイル内にオブジェクトが何も存在しない場合は、タイルの開始と終了を示すコマンドだけが記述される。

【 0 1 4 5 】

C 9 は図 8 のタイル B の描画コマンドの開始を示すコマンドである。C 1 0 は文字のフォントの種類を選択するためのコマンドであり、この場合には「 1 」という番号の付けられたフォントセットに設定されている。C 1 1 はフォントの大きさを設定するコマンドであり、この場合には「 1 0 ポイント」の大きさに設定されている。

【 0 1 4 6 】

C 1 2 は文字の色を設定するコマンドであり、順に R (赤) , G (緑) , B (青) 各色成分の輝度を示してある。この輝度は、例えば、0 から 2 5 5 の 2 5 6 段階で量子化されているものとする。この場合は、{ 0 , 0 , 0 } に設定されている。C 1 3 は文字を描画する開始位置の座標を示すコマンドである。座標位置 (X , Y) は、タイルの左上を原点に指定する。この場合は、タイルの { 0 , 5 } の位置から文字の描画を開始するように設定されている。C 1 4 は実際に描画する文字列 (X X X X) を示すコマンドである。C 1 5 はタイル B の描画コマンドの終了を示すコマンドである。

【 0 1 4 7 】

C 1 0 0 は図 8 のタイル C の描画コマンドの開始を示すコマンドである。C 1 0 1 は図形描画の際の面の塗りつぶしの色を示すコマンドである。色の指定は、文字の色と同様である。C 1 0 2 は図形描画の線の色を指定するコマンドである。C 1 0 3 は図形を描画する位置の座標を示すコマンドである。

【 0 1 4 8 】

C 1 0 4 は円弧を描画する際の半径を指定するコマンドであり、この場合、「 1 0 」座標単位を表している。C 1 0 5 は閉円弧の描画をするコマンドである。コマンド内の 2 個のパラメータは円弧を描画する際の描画開始角度と終了角度を示している。垂直情報を 0 度として、この場合には 0 度から 9 0 度の円弧を描画することを示している。C 1 0 6 はタイル C の描画コマンドの終了を示すコマンドである。

【 0 1 4 9 】

C 1 2 0 は図 8 のタイル D の描画コマンドの開始を示すコマンドである。C 1 2 1 ~ C 1 3 0 は、C 9 ~ C 1 5 のコマンドと同様に文字の描画命令による文字のフォントの種類、色、大きさ等を指定するコマンド、及び C 1 0 0 ~ C 1 0 6 までのコマンドと同様に図形の描画命令による図形の面、線の色、位置等を指定するコマンドである。C 1 3 1 はタイル D の描画コマンドの終了を示すコマンドである。

【 0 1 5 0 】

C 5 0 0 はページの終了を示すコマンドである。

10

20

30

40

50

【 0 1 5 1 】

次に、タイル / ページベクタ変換部 1 3 の処理の詳細について、図 1 0 を用いて説明する。

【 0 1 5 2 】

図 1 0 は本発明の実施形態 1 のタイル / ページベクタ変換部が実行する処理を示すフローチャートである。

【 0 1 5 3 】

尚、このタイル / ページベクタ変換部 1 3 では、ページベクタデータとタイルベクタデータ間での相互変換を行うことができる。あるいは、タイル / ページベクタ変換部 1 3 は、ページベクタデータからタイルベクタデータへ変換する変換部と、タイルベクタデータからページベクタデータへ変換する変換部で構成されていても良い。

10

【 0 1 5 4 】

(ステップ S 6 0 1)

まず、システムメモリ 5 中に記憶されたドキュメントデータ (ベクタデータ) から、ヘッダ部分に相当するコマンド列を読み込み、処理対象となるドキュメントデータ全体に関するコマンド部分を解析する。具体的には、図 7 または図 9 の C 1 ~ C 5 に相当する部分の内容を解析する。

【 0 1 5 5 】

(ステップ S 6 0 2)

解析の結果に基づいて、ドキュメントデータのタイプが、ページベクタデータであるか否かを判定する。ページベクタデータである場合 (ステップ S 6 0 2 で Y E S)、ステップ S 6 0 3 以降のステップに進み、ページベクタ タイルベクタ変換を実行する。一方、ページベクタデータでない場合、つまり、タイルベクタデータである場合 (ステップ S 6 0 2 で N O)、ステップ S 6 1 0 以降のステップに進み、タイルベクタ ページベクタ変換を実行する。

20

【 0 1 5 6 】

(ステップ S 6 0 3)

ページベクタデータから、オブジェクトを記述するコマンド列を読み込む。

【 0 1 5 7 】

(ステップ S 6 0 4)

ステップ S 6 0 3 で読み込んだコマンド列を解析し、記述されているオブジェクトの大きさが分割したいタイルサイズを超えているか否かを判定する。つまり、オブジェクトの更なる分割が必要であるか否かを判定する。

30

【 0 1 5 8 】

オブジェクトの大きさが分割したいタイルサイズを超えていない場合 (ステップ S 6 0 4 で N O)、ステップ S 6 0 5 をスキップして、ステップ S 6 0 6 に進む。一方、オブジェクトの大きさが分割したいタイルサイズを超えている場合 (ステップ S 6 0 4 で Y E S)、ステップ S 6 0 5 に進む。

【 0 1 5 9 】

(ステップ S 6 0 5)

ここでは、入力されたオブジェクトの分割処理を実行する。

40

【 0 1 6 0 】

例えば、図 7 のページベクタデータでは、文字の描画命令部分 9 0 2 で、「 X X X X ・ ・ ・ Y Y ・ ・ ・ 」を含む全ての文字列の描画命令を記述している。これに対して、図 9 のタイルベクタデータでは、例えば、タイル B に対する描画命令 1 1 0 4 では、「 X X X X 」の文字列の描画命令しか記述されていない。

【 0 1 6 1 】

従って、タイルベクタデータでは、文字列が複数のタイルに跨がる場合は、その文字列を途中 (タイル境界) で分割し、分割された後続の文字列は別の文字列として、次のタイルに記述を行う。次のタイルに記述が収まらなかったら、同様にまたタイルに含まれる文

50

字列を分割し、分割された全ての文字列がタイルサイズに収まるまで繰り返す。文字列をどこで切るかは、フォントの種類、サイズからタイル内に収まる文字数を算出し、その数だけの文字を抽出する。

【 0 1 6 2 】

例えば、図 7 のページベクタデータの文字の描画命令部分 9 0 2 に対して、タイル内に収まる文字数は 4 つとして算出され、図 7 のコマンド C 1 1 の記述は、タイル B に対する描画命令を構成するコマンド C 1 3 の記述に変換される。

【 0 1 6 3 】

また、図形の描画命令部分 9 0 3 では、コマンド C 1 7 ~ C 2 1 で記述されている図形（図 6 の 3 / 4 円）は、図 8 のタイルベクタデータを構成するタイルの 1 つには収まらない。そのため、この図形は、タイル D を含む複数のタイルに分割される。図形の分割は、図形の描画位置や図形の形、大きさからタイルの境界領域と接する部分を算出し、その境界とタイル内に収まっている図形の部分領域で構成される閉領域を新たな図形として記述し直す。

【 0 1 6 4 】

図 7 の図形の描画命令部分 9 0 3 で記述される図形（図 6 の 3 / 4 円）は、図 9 では、その左下の部分領域が、図形の描画命令部分 1 1 0 8 のコマンド C 1 2 6 から C 1 3 0 のような 1 / 4 円の記述に変換される。また、残りの領域も、同様な形の 1 / 4 円の記述に変換される。

【 0 1 6 5 】

（ステップ S 6 0 6 ）

入力されたオブジェクトのコマンドの記述に対し、タイルベクタデータ内での描画位置へ変更するために、座標位置の変換を行う。ページベクタデータでは、ページの左上からの位置を記述していたのに対し、タイルベクタデータではタイルの左上からの位置に記述し直す。描画位置をタイル内の座標で記述することにより、座標計算に要するデータ長を削減することが可能になる。

【 0 1 6 6 】

（ステップ S 6 0 7 ）

1 つのオブジェクトに対するコマンドの記述変換が終了したら、ページ内の全てのオブジェクトのコマンドの記述変換が終了したか否かを判定する。終了していない場合（ステップ S 6 0 7 で N O ） 、ステップ S 6 0 3 に戻り、次のコマンドに対して、ステップ S 6 0 3 ~ ステップ S 6 0 7 の処理を繰り返す。一方、終了した場合（ステップ S 6 0 7 で Y E S ） 、ステップ S 6 0 8 へ進む。

【 0 1 6 7 】

（ステップ S 6 0 8 ）

全てのコマンドの記述変換が終了したら、図 8 のように分割されたタイルベクタデータ中の各タイルに対し、ページの左上から順にタイルベクタデータとしてシステムメモリ 5 への書出を実行する。タイルベクタデータとしては、ステップ S 6 0 5 及びステップ S 6 0 6 で記述されたコマンドに対し、タイルの開始と終了を示すコマンドを追加するようなフォーマットで記述される。

【 0 1 6 8 】

まず、ページの一番最初のコマンドの書出時点では、システムメモリ 5 内にオブジェクトがない状態のタイルベクタを生成しておく。オブジェクトがないタイルベクタとしては、例えば、図 8 のタイル A がある。このタイル A は、タイルの開始と終了を示すコマンド C 7 及び C 8 のみから構成される描画命令部分 1 1 0 3 （図 9 ）で記述される。

【 0 1 6 9 】

次に、ステップ S 6 0 3 ~ ステップ S 6 0 7 で処理されたコマンドが存在する座標のタイルに対し、オブジェクトの記述を追加する。例えば、図 8 のタイル B の場合は、コマンド C 9 ~ C 1 5 から構成される描画命令部分 1 1 0 4 （図 9 ）で記述される。また、タイル D のように、同じタイルに複数のオブジェクトが存在する場合には、描画命令部分 1 1

10

20

30

40

50

06のように、文字の描画命令部分1107を構成するオブジェクト記述と、図形の描画命令部分1108を構成するオブジェクト記述を列記する。

【0170】

(ステップS609)

1つのオブジェクトのタイルベクタへの書出が終了すると、そのページのオブジェクトの記述が全て終了したか否かを判定する。終了していない場合(ステップS609でNO)、ステップS603に戻る。一方、終了した場合(ステップS609でYES)、処理を終了する。

【0171】

一方、ステップS602で、ドキュメントデータのタイプが、タイルベクタデータである場合について説明する。

10

【0172】

(ステップS610)

タイルベクタデータから、オブジェクトを記述するコマンド列を読み込む。

【0173】

(ステップS611)

ステップS610で読み込んだコマンド列を解析し、記述されているオブジェクトがそれより以前に読み込まれたタイルと結合可能であるか否かを判定する。結合可能でない場合(ステップS611でNO)、ステップS612をスキップし、ステップS613に進む。一方、結合可能である場合(ステップS611でYES)、ステップS612に進む。

20

【0174】

尚、オブジェクトが結合可能であるか否かの判定は、読み込んだコマンドの座標位置や、図形の種別等を基に判定する。また、文字列の場合は、フォントサイズやフォントの種類を基に判定する。

【0175】

(ステップS612)

オブジェクトの結合処理を実行する。この処理は、基本的には、ステップS605の処理手順を逆にすることで実現する。

【0176】

(ステップS613)

入力されたオブジェクトのコマンドの記述に対し、ページベクタデータ内での描画位置へ変更するために、座標位置の変換を行う。タイルベクタデータでは、タイルの左上からの位置を記述していたのに対し、ページベクタデータでは、ページの左上からの位置に記述し直す。

30

【0177】

(ステップS614)

1つのオブジェクトに対するコマンドの記述変換が終了したら、タイル内の全てのオブジェクトのコマンドの記述変換が終了したか否かを判定する。終了していない場合(ステップS614でNO)、ステップS610に戻り、次のコマンドに対して、ステップS610～ステップS613の処理を繰り返す。一方、終了した場合(ステップS614でYES)、ステップS615へ進む。

40

【0178】

(ステップS615)

全てのコマンドの記述変換が終了したら、ページベクタデータとしてシステムメモリ5への書出を実行する。ページベクタデータとしては、ステップS612及びステップS613で記述されたコマンドに対し、タイルの開始と終了を示すコマンドを削除したようなフォーマットで記述される。

【0179】

まず、ページ内の一番最初のタイルに記述されたコマンドの書出時点では、システムメ

50

メモリ 5 内にオブジェクトがない状態のページベクタを生成しておく。これは、図 7 の記述を用いて説明すると、コマンド C 1 ~ C 6、コマンド C 2 2 だけで記述されるページベクタである。

【 0 1 8 0 】

次に、ステップ S 6 1 0 ~ ステップ S 6 1 3 で処理されたオブジェクトの記述を追加する。図 7 の記述の場合、文字の描画命令部分 9 0 2 を構成するコマンド C 7 ~ C 1 1 がそれにあたる。この場合のオブジェクトは、文字列 { X X X X . . . Y Y . . . } を示しているが、これは、ステップ S 6 1 2 で、図 9 の文字の描画命令部分 1 1 0 4 や 1 1 0 7 等で記述される各タイル中の文字列を順次結合したオブジェクトの記述である。

【 0 1 8 1 】

(ステップ S 6 1 6)

1 つコマンドのページベクタへの書出が終了すると、そのタイルのオブジェクトの記述が全て終了したか否かを判定する。終了していない場合 (ステップ S 6 1 6 で N O)、ステップ S 6 1 0 に戻る。一方、終了した場合 (ステップ S 6 1 6 で Y E S)、ステップ S 6 1 7 に進む。

【 0 1 8 2 】

(ステップ S 6 1 7)

1 つのタイルベクタデータの書出が終了すると、そのページのタイルベクタデータの記述に対する処理が全て終了したか否かを判定する。終了していない場合 (ステップ S 6 1 7 で N O)、ステップ S 6 1 0 に戻る。一方、終了した場合 (ステップ S 6 1 7 で Y E S)、処理を終了する。

【 0 1 8 3 】

[画像データ展開部 (R I P)]

次に、コントローラ 1 中の画像データ展開部 1 8 の詳細について説明する。

【 0 1 8 4 】

まず、コピーやプリント、送信等の画像データに対する処理を開始する前に、ローカルメモリ 1 9 の初期化と、作成するオブジェクトの解像度の設定を行っておく。実施形態 1 では、生成解像度は 6 0 0 d p i とし、ポイントサイズや m m 等の単位系で指定された印刷コマンドは、この値を用いてドット数に変換されることになる。

【 0 1 8 5 】

以下では、画像データ展開部 1 8 によって、タイルベクタデータ展開を実行する場合の処理について、図 1 1 を用いて説明する。

【 0 1 8 6 】

図 1 1 は本発明の実施形態 1 の画像データ展開部が実行する処理を示すフローチャートである。

【 0 1 8 7 】

(ステップ S 7 1)

システムメモリ 5 より、S B B 2 を経由して、R I P 1 8 に一定サイズ分入力されたタイルベクタデータは、ローカルメモリ 1 9 のタイルベクタ領域に一時的に格納する。

【 0 1 8 8 】

(ステップ S 7 2)

タイルベクタデータをローカルメモリ 1 9 に格納したら、R I P 1 8 内の μ R I P 1 8 a ~ 1 8 d のいずれかが、タイルベクタデータを展開可能 (処理可能) であるか否かを判定する。 μ R I P 1 8 a ~ 1 8 d のいずれもタイルベクタデータの展開 (処理) 中である場合 (ステップ S 7 2 で N O)、それらのいずれかが展開可能になるまで待機する。

【 0 1 8 9 】

(ステップ S 7 3)

μ R I P 1 8 a ~ 1 8 d のいずれかがタイルベクタデータを展開可能になれば、予め定められた文法に従って、ローカルメモリ 1 9 に格納されたタイルベクタデータのコマンド解析を実行する。

【 0 1 9 0 】

(ステップ S 7 4)

コマンド解析結果に基づいて、コマンドが描画命令であるか排紙命令であるかを判定する。描画命令である場合 (ステップ S 7 4 で Y E S)、ステップ S 7 5 に進む。一方、排紙命令である場合 (ステップ S 7 4 で N O)、ステップ S 7 6 に進む。

【 0 1 9 1 】

(ステップ S 7 5)

コマンドが描画命令である場合、描画オブジェクト (D L データ) の生成を実行する。タイルベクタデータ内のコマンドが文字の描画命令である場合、そのコマンドで指定されるフォントの書体や、文字サイズ、文字コードを元にフォントオブジェクトを生成して、ローカルメモリ 1 9 の D L データ領域に格納する。また、文字以外の描画命令、つまり、図形の描画命令である場合、そのコマンドで指定された図形 (ラインや円、多角形等) の描画オブジェクトを生成し、ローカルメモリ 1 9 の D L データ領域に格納する。

10

【 0 1 9 2 】

また、コマンドが描画命令で指定されない印刷データである場合には、この印刷データに応じて印刷位置移動や印刷環境設定等の印刷制御処理を実行し、一単位分のコマンド解釈を終了する。

【 0 1 9 3 】

これは、タイルベクタデータ内のコマンド全ての解釈が終了するまで、上記処理を繰り返す。

20

【 0 1 9 4 】

(ステップ S 7 6)

コマンドが排紙命令である場合、μ R I P は、ローカルメモリ 1 9 上のタイルラスト領域に空き領域があるか否かを判定する。空き領域がない場合 (ステップ S 7 6 で N O)、他の μ R I P の処理が終了して、タイルラスト領域が解放されて、空き領域ができるまで待機する。一方、空き領域がある場合 (ステップ S 7 6 で Y E S)、ステップ S 7 7 に進む。

【 0 1 9 5 】

(ステップ S 7 7)

タイルラスト領域に空き領域がある場合、ステップ S 7 5 で生成された描画オブジェクトを読み出し、タイルラスト領域に描画 (ラスタライズ) する。この時、生成解像度が 6 0 0 d p i であれば、タイルラスト領域には 6 0 0 d p i の画像としてラスタライズされる。そして、描画が終了したタイルラスト画像は、S B B 2 を介して、画像処理部 1 5 に出力する。

30

【 0 1 9 6 】

(ステップ S 7 8)

ステップ S 7 5 またはステップ S 7 7 で一つのタイルベクタデータに対する、コマンド解析または描画処理が終了したら、タイルベクタ領域に格納したタイルベクタデータのすべてに対して処理が終了したか否かを判定する。未処理のタイルベクタデータがある場合 (ステップ S 7 8 で N O)、ステップ S 7 2 に戻り、次のタイルベクタデータの処理を続ける。一方、未処理のタイルベクタデータがない場合 (ステップ S 7 8 で Y E S)、ステップ S 7 9 に進む。

40

【 0 1 9 7 】

(ステップ S 7 9)

1 ページ分のタイルベクタデータに対し、全ての処理が終了したか否かを判定する。未処理のタイルベクタデータがある場合 (ステップ S 7 9 で N O)、ステップ S 7 1 に戻り、システムメモリ 5 から、タイルベクタデータを読み出して、処理を続ける。一方、未処理のタイルベクタデータがない場合 (ステップ S 7 9 で Y E S)、処理を終了する。

【 0 1 9 8 】

以上説明したように、実施形態 1 によれば、システムメモリ上に、入力された画像デー

50

タをページベクタデータとタイルベクタデータの2種類の形式の画像データのみを記憶する構成を採用する。これにより、画像サイズの大きいラスタデータ及びDLデータをシステムメモリに記憶する必要がなくなる。そのため、システムメモリ上で確保しなければならない画像データ領域を削減することができる。

【0199】

また、外部機器に画像データを送信する際にはタイルベクタデータをページベクタデータに変換して、画像データ内に存在するオブジェクト数を減らすことで、送信データ量を削減することができる。また、送信先への画像データをページベクタデータで送信することで、送信先では、受信した画像データ(ページベクタデータ)を、PDFやSVG等の汎用フォーマットへ容易に変換することができる。

10

【0200】

また、データプールをベクタデータで行うことにより、システムの解像度依存性が解消され、画質とシステムの処理速度向上が並行して実現できるようになる。つまり、コストパフォーマンスの高いシステムを構成することができる。

【0201】

さらに、システム内の画像データをベクタデータで取り扱うことにより、蓄積画像データの検索性の向上や、ラスタデータをオブジェクト化して再利用するようなPOD印刷に対応するための画像加工を高速化に寄与する等の、よりインテリジェントなシステムを実現することができる。

【0202】

20

また、ベクタデータに展開するための画像データ展開部を複数個用意して並列に動作させることで、タイルベクタデータの展開を高速化することができる。また、動作させる画像データ展開部の個数を制御することで、用途や目的に応じて、その処理機能を可変にできるスケーラブルなシステムを構築することが可能となる。

【0203】

[実施形態2]

実施形態1では、MFPから外部機器への送信を行う際には、タイルベクタデータをページベクタデータに変換して、データ量の削減や画像データの汎用性を高めているが、これに限定されない。例えば、画像データを記憶する際にタイルベクタデータだけではなく、予めページベクタデータも併せて記憶するようにしても良い。このような構成の場合、送信の際に、タイルベクタデータからページベクタデータへ変換することなく、外部機器にページベクタデータを送信することもできる。

30

【0204】

尚、システム全体の構成は、実施形態1と同様なので、その詳細については割愛する以下、実施形態2における画像処理システムでの送信動作について説明する。

【0205】

[送信]

図12は本発明の実施形態2の画像処理システムでの送信動作に係るデータフローを示す図である。

【0206】

40

尚、このデータフローは、CPU3の制御の下、MFP1000を構成する各種構成要素を協調動作させることで実現される。

【0207】

(S131): 実施形態2では、1つの画像データに対し、HDD8にはタイルベクタデータとページベクタデータとが関連付けて記憶される。送信時は、その内のページベクタデータが選択され、SBB2を介して、汎用バス6に接続されたHDDCont7より読み出され、システムメモリ5に一時的に記憶される。

【0208】

(S132): システムメモリ5に記憶されたページベクタデータは、汎用バス6に接続されたネットワークI/F11から読み出され、ネットワーク12に接続された外部機

50

器へと転送される。

【0209】

尚、実施形態2では、実施形態1に比べて、HDD8に記憶すべき画像データ容量が多くなるが、送信時のフローが非常に簡略化される。

【0210】

以上説明したように、実施形態2によれば、実施形態1で説明した効果に加えて、1つの画像データに対して、タイルベクタデータと、それに対応するページベクタデータを併せてHDDに記憶管理しておく。これにより、記憶管理されている2種類のベクタデータをHDDから適宜選択するだけで、画像データの出力時により適切なベクタデータを、変換処理を行うことなく、出力することができる。

10

【0211】

[実施形態3]

実施形態3では、実施形態1の図10のステップS605のオブジェクトの分割処理の応用例について説明する。

【0212】

実施形態3では、タイル/ページベクタ変換部13において、特に、処理対象のオブジェクトが曲線オブジェクトである場合に、その曲線オブジェクトの分割処理について説明する。

【0213】

図13は本発明の実施形態3のタイルベクタデータの例を示す図である。

20

【0214】

図13では、ドキュメントデータ1401の短手方向を「X」方向、長手方向を「Y」と定義する。また、図中のX方向に配列された数列は、X方向のタイルID、Y方向に配列された数列は、Y方向のタイルIDを表す。A～Cは、それぞれタイルID=(0,0)、(2,3)、(3,3)の位置にあるタイルベクタを示している。

【0215】

特に、図13では、タイルB及びCに曲線オブジェクトが跨がっている、つまり、曲線オブジェクトがタイルB及びCによって分割されている例を示している。

【0216】

ここで、このドキュメントデータ1401がタイル分割前のページベクタデータである場合、その内容を構成する記述例について、図14を用いて説明する。

30

【0217】

図14は本発明の実施形態3のページベクタデータの記述例を示す図である。

【0218】

図14において、1501はドキュメントデータ全体の設定に関わるドキュメント設定命令部分、1502は描画命令部分全体を示している。

【0219】

各描画命令の詳細について説明する。

【0220】

ドキュメント設定命令部分1501において、C1～C5は、ドキュメント全体に関係するコマンドである。従って、これらのコマンドC1～C5は、ドキュメント1部について1ヶ所しか付いていない。

40

【0221】

これらドキュメントデータ全体に関係するコマンドには、実施形態1の図7で詳述した内容と同様である。

【0222】

描画命令部分を構成するC6～C10は、ドキュメントデータを出力するための各種コマンドである。

【0223】

C6はページの開始を示すコマンドである。C7は線(曲線)の色を設定するコマンド

50

であり、順に R (赤) , G (緑) , B (青) 各色成分の輝度を示してある。この輝度は、例えば、0 から 2 5 5 の 2 5 6 段階で量子化されているものとする。この場合は、{ 2 5 5 , 9 2 , 1 2 8 } に設定されている。

【 0 2 2 4 】

C 8 は曲線を描画する開始位置 (曲線を規定する一方の端点) の座標を示すコマンドである。座標位置 (X , Y) は、ページの左上を原点に指定する。この場合は、ページの { 6 6 , 1 1 2 } の位置から曲線の描画を開始するように設定されている。C 9 は曲線を描画するアンカーポイントと終了位置 (曲線を規定する他方の端点) の座標を示すコマンドである。この場合は、ページの { 1 2 6 , 9 8 } 及び { 6 6 , 1 2 6 } のアンカーポイントを通り、{ 1 2 6 , 1 1 2 } の位置で曲線の描画を終了するように設定されている。C 1 0 はページの終了を示すコマンドである。

10

【 0 2 2 5 】

これに対して、ドキュメントデータ 1 4 0 1 がタイル分割後のタイルベクタデータである場合、その内容を構成する記述例について、図 1 5 を用いて説明する。

【 0 2 2 6 】

図 1 5 は本発明の実施形態 3 のタイルベクタデータの記述例を示す図である。

【 0 2 2 7 】

図 1 5 において、1 6 0 1 はドキュメントデータ全体の設定に関わるドキュメント設定命令部分、1 6 0 2 は描画命令部分全体、1 6 0 3 ~ 1 6 0 5 はタイル A ~ C の描画命令部分を示している。1 6 0 6 及び 1 6 0 7 はタイル B の図形 (曲線) の描画命令部分、1 6 0 8 及び 1 6 0 9 はタイル C の図形 (曲線) の描画命令部分を示している。

20

【 0 2 2 8 】

各描画命令の詳細について説明する。

【 0 2 2 9 】

ドキュメント設定命令部分 1 6 0 1 において、C 1 ~ C 5 は、ドキュメント全体に関するコマンドである。従って、これらのコマンド C 1 ~ C 5 は、ドキュメント 1 部について 1 ケ所しか付いていない。

【 0 2 3 0 】

これらドキュメントデータ全体に関するコマンドは、実施形態 1 の図 9 で詳述した内容と同様である。

30

【 0 2 3 1 】

描画命令部分 1 6 0 2 を構成する C 6 ~ C 5 0 0 は、ドキュメントデータを出力するための各種コマンドである。

【 0 2 3 2 】

C 6 はページの開始を示すコマンドである。C 7 は図 1 3 のタイル A の描画コマンドの開始を示すコマンドである。ここで、T i l e S t a r t (0 , 0) 中の 2 個のパラメータは、ドキュメントデータ内におけるタイルの I D を示している。C 8 はタイル A の描画コマンドの終了を示すコマンドである。タイル A のように、そのタイル内にオブジェクトが何も存在しない場合は、タイルの開始と終了を示すコマンドだけが記述される。

【 0 2 3 3 】

40

ここで、タイル B 及び C については、ページベクタデータ上の 1 つの曲線オブジェクトが、タイル B 及び C によって 4 つの曲線オブジェクトに分割される。そして、その 4 つの曲線オブジェクトの内、2 つの曲線オブジェクトがそれぞれタイル B 及び C に属する。

【 0 2 3 4 】

そのため、タイル B の描画命令部分 1 6 0 4 では、その 2 つの曲線オブジェクトを構成する色としてコマンド C 1 0 1、各曲線オブジェクトの開始位置、アンカーポイント、終了位置からなる描画命令部分 1 6 0 6 (コマンド C 1 0 2 及び C 1 0 3)、描画命令部分 1 6 0 7 (コマンド C 1 0 4 及び C 1 0 5) が記述されている。

【 0 2 3 5 】

同様に、タイル C の描画命令部分 1 6 0 5 では、その 2 つの曲線オブジェクトを構成す

50

る色としてコマンド C 1 2 1、各曲線オブジェクトの開始位置、アンカーポイント、終了位置からなる描画命令部分 1 6 0 8 (コマンド C 1 2 2 及び C 1 2 3)、描画命令部分 1 6 0 9 (コマンド C 1 2 4 及び C 1 2 5) が記述されている。

【 0 2 3 6 】

次に、タイル B 及びタイル C に跨がる曲線オブジェクトのタイル分割について、図 1 6 を用いて説明する。

【 0 2 3 7 】

図 1 6 は本発明の実施形態 3 の曲線オブジェクトのタイル分割を説明するための図である。

【 0 2 3 8 】

図 1 6 (a) に示す 1 つ以上のタイルに跨がる曲線オブジェクトは、図 1 6 (b) に示すように、まず、その曲線が内包される凸多角形 P 0 - P 1 - P 3 - P 2 を算出する。次に、この曲線オブジェクトが注目タイルのどの辺と交差している可能性があるかを、タイル辺 1 6 A - 1 6 B、1 6 B - 1 6 D、1 6 D - 1 6 C、1 6 C - 1 6 A と、線分 P 0 - P 1、P 1 - P 3、P 3 - P 2、P 2 - P 1 のそれぞれの組み合わせで交差するか否かを評価する。

【 0 2 3 9 】

ここでは、図 1 6 (c) に示すように、線分 P 0 - P 1 及び線分 P 3 - P 2 と交差したタイル辺 1 6 B - 1 6 D について、曲線 P 0 - P 3 の交点を算出し、交点 P 4、P 5、P 6 を取得する。次に、曲線 P 0 - P 3 を、交点 P 4、P 5、P 6 を端点とした曲線 P 0 - P 4、P 4 - P 5、P 5 - P 6、P 6 - P 3 に分割する。更に、それぞれの曲線のアンカーポイント P 7、P 8、P 9、P 10 等を算出し、分割された曲線を描画するためのコマンドを生成する。

【 0 2 4 0 】

以上のようにして、1 つ以上のタイルに跨がる曲線オブジェクトを各タイルに分割することができる。

【 0 2 4 1 】

次に、曲線オブジェクトの分割処理の詳細について、図 1 7 を用いて説明する。

【 0 2 4 2 】

図 1 7 は本発明の実施形態 3 の曲線オブジェクトの分割処理の詳細を示すフローチャートである。

【 0 2 4 3 】

尚、この処理は、図 6 のステップ S 6 0 5 のオブジェクトの分割処理において、処理対象が曲線オブジェクトである場合に実行する。

【 0 2 4 4 】

(ステップ S 1 8 0 1)

まず、処理対象の曲線オブジェクトが、既に他の隣接タイルで分割されているか否かを判定する。分割されている場合 (ステップ S 1 8 0 1 で YES)、ステップ S 1 8 0 9 に進む。一方、分割されていない場合 (ステップ S 1 8 0 1 で NO)、ステップ S 1 8 0 2 に進む。

【 0 2 4 5 】

(ステップ S 1 8 0 2)

処理対象の曲線オブジェクトを内包する凸多角形とその頂点を算出する。

【 0 2 4 6 】

(ステップ S 1 8 0 3)

処理対象の曲線オブジェクトを内包する凸多角形が、注目タイルのどの辺と交差可能性があるかを評価する。

【 0 2 4 7 】

(ステップ S 1 8 0 4)

ステップ S 1 8 0 3 の評価結果から、処理対象の曲線オブジェクトを内包する凸多角形

10

20

30

40

50

が注目タイルのどれかの辺と交差可能性があるか否かを判定する。交差可能性がない場合（ステップS1804でNO）、処理を終了する。一方、交差可能性がある場合（ステップS1804でYES）、ステップS1805に進む。

【0248】

（ステップS1805）

処理対象の曲線オブジェクトを内包する凸多角形と交差可能性のある注目タイルの注目辺（タイル辺）を選択し、当該タイル辺とその曲線オブジェクトの交点を算出する。また、その算出した交点をシステムメモリ5に記憶する。ここで、交点数は、例えば、0から3の可能性はある。

【0249】

10

尚、この交点の算出は、例えば、凸多角形と構成する注目タイルのタイル辺に基づいて規定される方程式に基づいて算出する。

【0250】

（ステップS1806）

処理対象の曲線オブジェクトを内包する凸多角形と交差可能性のある注目辺との交点をすべて算出したか否かを判定する。未算出の注目辺の交点がある場合（ステップS1806でNO）、ステップS1805に戻る。一方、未算出の注目辺の交点がない場合（ステップS1806でYES）、ステップS1807に進む。

【0251】

ここで、ステップS1802～ステップS1804の処理を実行することで、処理量の多い交点算出処理（ステップS1805及びステップS1806）のループ回数を低減することができる。

20

【0252】

（ステップS1807）

注目タイルの注目辺と曲線オブジェクトの注目交点を用いて、曲線オブジェクトを分割する。また、分割された曲線の端点とアンカーポイントを算出して、システムメモリ5に記憶する。

【0253】

（ステップS1808）

システムメモリ5に記憶されているタイル辺と曲線オブジェクトの交点すべてにおいて、曲線オブジェクトの分割が終了したか否かを判定する。分割が終了していない場合（ステップS1808でNO）、ステップS1807に戻る。一方、分割が終了している場合（ステップS1808でYES）、処理を終了する。

30

【0254】

（ステップS1809）

一方、ステップS1801において、処理対象の曲線オブジェクトが、既に他の隣接タイルで分割されている場合、その他の隣接タイルで処理されたときにシステムメモリ5に記憶された、その曲線オブジェクトと隣接タイルとの交点を取得する。

【0255】

（ステップS1810）

他の隣接タイルで算出した、曲線オブジェクトを内包する凸多角形とその頂点を取得して、ステップS1803へ進む。

40

【0256】

次に、ステップS1802の処理の詳細について、図18を用いて説明する。

【0257】

図18は本発明の実施形態3のステップS1802の処理の詳細を示すフローチャートである。

【0258】

（ステップS1901）

曲線オブジェクトの端点及びアンカーポイントの4点で形成された四角形P0 - P1 -

50

P 3 - P 2 (例えば、図 1 6 (b)) の一頂点 P 0 が、他の頂点からなる三角形 P 1 - P 2 - P 3 に内包されるか否かを判定する。内包される場合 (ステップ S 1 9 0 1 で Y E S)、ステップ S 1 9 0 6 へ進む。一方、内包されない場合 (ステップ S 1 9 0 1 で N O)、ステップ S 1 9 0 2 へ進む。

【 0 2 5 9 】

(ステップ S 1 9 0 2)

曲線オブジェクトの端点及びアンカーポイントの 4 点で形成された四角形 P 0 - P 1 - P 3 - P 2 の一頂点 P 1 が、他の頂点からなる三角形 P 0 - P 2 - P 3 に内包されるか否かを判定する。内包される場合 (ステップ S 1 9 0 2 で Y E S)、ステップ S 1 9 0 7 へ進む。一方、内包されない場合 (ステップ S 1 9 0 2 で N O)、ステップ S 1 9 0 3 へ進む。

10

【 0 2 6 0 】

(ステップ S 1 9 0 3)

曲線オブジェクトの端点及びアンカーポイントの 4 点で形成された四角形 P 0 - P 1 - P 3 - P 2 の一頂点 P 2 が、他の頂点からなる三角形 P 0 - P 1 - P 3 に内包されるか否かを判定する。内包される場合 (ステップ S 1 9 0 3 で Y E S)、ステップ S 1 9 0 8 へ進む。一方、内包されない場合 (ステップ S 1 9 0 3 で N O)、ステップ S 1 9 0 4 へ進む。

【 0 2 6 1 】

(ステップ S 1 9 0 4)

曲線オブジェクトの端点及びアンカーポイントの 4 点で形成された四角形 P 0 - P 1 - P 3 - P 2 の一頂点 P 3 が、他の頂点からなる三角形 P 0 - P 1 - P 2 に内包されるか否かを判定する。内包される場合 (ステップ S 1 9 0 4 で Y E S)、ステップ S 1 9 0 9 へ進む。一方、内包されない場合 (ステップ S 1 9 0 4 で N O)、ステップ S 1 9 0 5 へ進む。

20

【 0 2 6 2 】

(ステップ S 1 9 0 5)

四角形 P 0 - P 1 - P 3 - P 2 を、曲線オブジェクトを内包する凸多角形とし、これの頂点として P 0 , P 1 , P 2 , P 3 をシステムメモリ 5 に記憶し、処理を終了する。

【 0 2 6 3 】

(ステップ S 1 9 0 6)

三角形 P 1 - P 2 - P 3 を、曲線オブジェクトを内包する凸多角形とし、これの頂点として P 1 , P 2 , P 3 をシステムメモリ 5 に記憶し、処理を終了する。

30

【 0 2 6 4 】

(ステップ S 1 9 0 7)

三角形 P 0 - P 2 - P 3 を、曲線オブジェクトを内包する凸多角形とし、これの頂点として P 0 , P 2 , P 3 をシステムメモリ 5 に記憶し、処理を終了する。

【 0 2 6 5 】

(ステップ S 1 9 0 8)

三角形 P 0 - P 1 - P 3 を、曲線オブジェクトを内包する凸多角形とし、これの頂点として P 0 , P 1 , P 3 をシステムメモリ 5 に記憶し、処理を終了する。

40

【 0 2 6 6 】

(ステップ S 1 9 0 9)

三角形 P 0 - P 1 - P 2 を、曲線オブジェクトを内包する凸多角形とし、これの頂点として P 0 , P 1 , P 2 をシステムメモリ 5 に記憶し、処理を終了する。

【 0 2 6 7 】

次に、ステップ S 1 8 0 3 の処理の詳細について、図 1 9 を用いて説明する。

【 0 2 6 8 】

図 1 9 は本発明の実施形態 3 のステップ S 1 8 0 3 の処理の詳細を示すフローチャートである。

50

【 0 2 6 9 】

(ステップ S 2 0 0 1)

まず、注目タイルが曲線オブジェクトを内包する凸多角形に内包されるか否かを判定する。内包される場合 (ステップ S 2 0 0 1 で Y E S)、ステップ S 2 0 0 2 へ進む。一方、内包されない場合 (ステップ S 2 0 0 1 で N O)、ステップ S 2 0 0 3 へ進む。

【 0 2 7 0 】

(ステップ S 2 0 0 2)

すべてのタイル辺が曲線オブジェクトと交差可能性があることを示す情報をシステムメモリ 5 に記憶し、処理を終了する。

【 0 2 7 1 】

(ステップ S 2 0 0 3)

曲線オブジェクトを内包する凸多角形がタイルに内包されるか否かを判定する。内包される場合 (ステップ S 2 0 0 3 で Y E S)、ステップ S 2 0 0 4 へ進む。一方、内包されない場合 (ステップ S 2 0 0 3 で N O)、ステップ S 2 0 0 5 へ進む。

【 0 2 7 2 】

(ステップ S 2 0 0 4)

どのタイル辺も曲線オブジェクトと交差可能性がないことを示す情報をシステムメモリ 5 に記憶し、処理を終了する。

【 0 2 7 3 】

(ステップ S 2 0 0 5)

4 つのタイル辺のひとつを注目タイル辺として設定する。

【 0 2 7 4 】

(ステップ S 2 0 0 6)

3 つもしくは 4 つの凸多角形の辺のひとつを注目凸多角形辺として設定する。

【 0 2 7 5 】

(ステップ S 2 0 0 7)

注目タイル辺と、注目凸多角形辺が交差するか否かを判定する。交差する場合 (ステップ S 2 0 0 7 で Y E S)、ステップ S 2 0 0 8 へ進む。一方、交差しない場合 (ステップ S 2 0 0 7 で N O)、ステップ S 2 0 0 9 へ進む。

【 0 2 7 6 】

(ステップ S 2 0 0 8)

注目タイル辺が曲線オブジェクトと交差可能性があることを示す情報をシステムメモリ 5 に記憶する。

【 0 2 7 7 】

(ステップ S 2 0 0 9)

3 つもしくは 4 つの凸多角形辺すべてに関して、注目凸多角形辺に対する設定処理 (ステップ S 2 0 0 6 及びステップ S 2 0 0 7) が完了しているか否かを判定する。完了していない場合 (ステップ S 2 0 0 9 で N O)、ステップ S 2 0 0 6 へ戻る。一方、完了している場合 (ステップ S 2 0 0 9 で Y E S)、ステップ S 2 0 1 0 へ進む。

【 0 2 7 8 】

(ステップ S 2 0 1 0)

4 つのタイル辺すべてに関して、注目タイル辺に対する設定処理 (ステップ S 2 0 0 5) が完了しているか否かを判定する。完了していない場合 (ステップ S 2 0 1 0 で N O)、ステップ S 2 0 0 5 へ戻る。一方、完了している場合 (ステップ S 2 0 1 0 で Y E S)、処理を終了する。

【 0 2 7 9 】

以上説明したように、実施形態 3 によれば、実施形態 1 及び 2 で説明した効果に加えて、曲線オブジェクトが存在するページベクタデータをタイルベクタデータに変換する際には、その曲線オブジェクトの形状を評価して、その評価結果に基づいて、タイルベクタデータに変換する。これにより、その曲線オブジェクトが適切にタイル上に分割されたタイ

10

20

30

40

50

ルベクタデータを生成することができる。

【0280】

[実施形態4]

図29で示す従来のシステムでは、RIP113でページベクタデータがラスタデータにレンダリングされる際、ページベクタデータに含まれるイメージオブジェクトは、レンダリングを行う解像度に応じて解像度変換がなされた上で、ラスタデータが生成される。

【0281】

イメージオブジェクトの解像度変換の際には、補間処理のないニアレストネイバー法で拡大縮小される場合と、エイリアシングの副作用を避けるためにバイリニア法やバイキュービック法等の補間処理が行われる場合がある。また、特許登録第3111971号公報

10

【0282】

一方、上記実施形態1では、イメージオブジェクトを含むページベクタデータをタイルベクタデータに変換する方法としては、例えば、以下のいくつかの方法が考えられる。

【0283】

まず、イメージオブジェクトを含むページベクタデータをタイルベクタデータに分割する際、単一のイメージオブジェクトが存在する複数のタイルに、イメージオブジェクトそのものをコピーして、タイルに埋め込む方法がある。この場合、イメージオブジェクトが存在するタイルN個それぞれに対してイメージオブジェクト全体を埋め込むと、イメージ

20

【0284】

一方、複数のタイルに跨がって存在するイメージオブジェクトをタイル境界で分割し、それぞれのタイルベクタデータに、それぞれ対応する分割後のイメージオブジェクトを埋め込む方法を採用すれば、タイルベクタデータ変換に伴うデータ量の増大は回避可能である。

【0285】

ここで、イメージオブジェクトをレンダリング後の解像度に合わせて解像度変換する際には、バイリニア法やバイキュービック法等の補間処理を選択することも可能である。しかしながら、この場合、タイル境界で分割されたイメージオブジェクトは、タイル境界外に画素を持たない。

30

【0286】

そのため、これらの補間処理を行うと、タイルベクタデータをレンダリングして、タイルラスタデータからなるページラスタデータを生成する際には、ページベクタデータ上のタイル境界に相当する位置にイメージオブジェクトの不連続領域が生じ、画質の劣化が発生することになる。

【0287】

そこで、上記の2種類の方法による課題を解決するために、イメージオブジェクトをタイル境界で分割する際には、そのイメージオブジェクトを構成する一部の画素を重複して分割し、補間処理に必要なタイル境界外の画素も併せてタイルベクタデータに埋め込む方法が考えられる。

40

【0288】

この方法を用いれば、データ増大の課題と、補間処理による画質劣化の課題を解決することが可能となる。しかしながら、この方法を採用したとしても、同一の画素を二重以上持つ必要が発生する。そのため、この方法でも、上記の方法ほどではないが、タイルベクタデータへの変換に伴ってデータ量が増大することになる。

【0289】

そこで、実施形態4では、特に、イメージオブジェクトを含むページベクタデータから

50

タイルベクタデータへ変換する際のデータ量増大を抑え、かつ画質劣化が生じるような補間処理を行わないことで、エイリアシングの発生も抑えることが可能な、イメージオブジェクトの分割処理について説明する。

【0290】

実施形態4では、イメージオブジェクトの解像度は、一般的に、300dpi以下であり、この解像度は、通常は、600dpi以上となるプリント解像度よりも低解像度である場合がほとんどである点に着目している。また、エイリアシングの問題は、画像を縮小する場合が、拡大する場合よりもより顕著な問題となる点に着目している。

【0291】

そこで、実施形態4では、閾値解像度を定義して、同一イメージオブジェクトを分割する際に、当該イメージオブジェクトの解像度が閾値解像度よりも高ければ、タイル境界外の周辺画素をベクタタイルデータに重複して保持する。一方、当該イメージオブジェクトの解像度が閾値解像度よりも低ければ、タイル境界外の周辺画素をベクタタイルデータに重複して保持しない構成を採用する。

10

【0292】

更に、イメージオブジェクトのレンダリング時には、タイル境界外周辺画素がない場合にはニアレストネイバー法によって解像度を変換し、タイル境界外周辺画素がある場合にはバイキュービック法、バイリニア法等の補間処理を用いて解像度を変換するように、その変換方法を適応的に切り替える。これにより、イメージオブジェクトのタイル分割によるデータ量増大を抑え、かつ補間処理を行わないことによるエイリアシングの発生も抑えることを実現する。

20

【0293】

以下、実施形態4では、実施形態1の図6のステップS605のオブジェクトの分割処理の応用例について説明する。

【0294】

実施形態4では、タイル/ページベクタ変換部13において、特に、処理対象のオブジェクトがイメージオブジェクトである場合の、そのイメージオブジェクトの分割処理について説明する。

【0295】

まず、1つ以上のタイルに跨がるイメージオブジェクトのタイル分割について、図20を用いて説明する。

30

【0296】

図20は本発明の実施形態4のイメージオブジェクトのタイル分割を説明するための図である。

【0297】

図20に示すイメージオブジェクト2000は、ページベクタデータに含まれており、タイルベクタデータに分割する際には、その配置された座標からタイル1402~1405に跨がって存在する。

【0298】

タイル/ページベクタ変換部13によるオブジェクトの分割処理では、まず、イメージオブジェクト2000の解像度と所定解像度(閾値解像度:例えば、600dpi)を比較する。

40

【0299】

尚、この所定解像度は、通常は、RIP18が生成するラスタデータの解像度であるとする。また、複数種類の解像度のラスタデータを生成可能である場合は、それらの内、最低解像度としても良い。

【0300】

比較の結果、イメージオブジェクト2000の解像度が所定解像度よりも低解像度である場合、タイル1402~1405に跨がって存在するイメージオブジェクト2000を、そのタイル1402~1405のタイル境界で分割する。そして、その分割されたイメ

50

ージオブジェクト1406～1409をそれぞれ対応するタイル1402～1405へ埋め込む。

【0301】

一方、イメージオブジェクト2000の解像度が所定解像度よりも高解像度である場合、タイル1402～1405に跨がって存在するイメージオブジェクト2000を、タイル1402～1405のタイル境界を越えるイメージオブジェクト2000の一部が互いに重複するように分割する。そして、その分割されたイメージオブジェクト1410～1413をそれぞれ対応するタイル1402～1405へ埋め込む。

【0302】

次に、イメージオブジェクトの分割処理の詳細について、図21を用いて説明する。

10

【0303】

図21は本発明の実施形態4のイメージオブジェクトの分割処理の詳細を示すフローチャートである。

【0304】

(ステップS1501)

まず、タイルベクタデータの生成対象となる注目タイルを選択する。

【0305】

(ステップS1502)

処理対象のイメージオブジェクトを規定するイメージ座標に注目タイルのタイル境界を投影する。

20

【0306】

(ステップS1503)

イメージ座標に投影されたタイル境界とイメージオブジェクト自身の外周からなる凸多角形を算出する。当該凸多角形に外接し、かつ直立した長方形を算出する。

【0307】

(ステップS1504)

イメージオブジェクトの解像度と所定解像度とを比較し、イメージオブジェクトの解像度が所定解像度より高いか否かを判定する。所定解像度以下である場合(ステップS1504でNO)、ステップS1505へ進む。一方、所定解像度より大きい場合(ステップS1504でYES)、ステップS1507へ進む。

30

【0308】

(ステップS1505)

凸多角形に外接する直立した長方形を、イメージオブジェクトを分割する境界として設定する。

【0309】

(ステップS1506)

分割後のイメージオブジェクトが、タイル境界外に画素を持たないことを示す情報をシステムメモリ5に記憶する。

【0310】

(ステップS1507)

一方、ステップS1504で、イメージオブジェクトの解像度が所定解像度より大きい場合、凸多角形に外接する直立した長方形を、補間処理の参照領域分拡大した拡大長方形を算出する。

40

【0311】

(ステップS1508)

拡大長方形を、イメージオブジェクトを分割する境界として設定する。

【0312】

(ステップS1509)

分割後のイメージオブジェクトが、タイル境界外に画素を持つことを示す情報をシステムメモリ5に記憶する。

50

【 0 3 1 3 】

(ステップ S 1 5 1 0)

設定された境界を外周とするイメージを、処理対象のイメージオブジェクトから抽出する。

【 0 3 1 4 】

(ステップ S 1 5 1 1)

抽出したイメージを、イメージオブジェクトとして注目タイルへ付与して、処理を終了する。

【 0 3 1 5 】

次に、図 2 1 のイメージオブジェクトの分割処理の具体例について、図 2 2 を用いて説明する。

10

【 0 3 1 6 】

図 2 2 は本発明の実施形態 4 のイメージオブジェクトの分割処理の具体例を示す図である。

【 0 3 1 7 】

イメージオブジェクト 2 2 0 1 は、独自のイメージ座標を持っている (図 2 2 (a)) 。そして、このイメージオブジェクト 2 2 0 1 は、ページベクタデータ 2 2 0 2 に対して、アフィン変換によってページベクタデータ座標に投影され (図 2 2 (b)) 、ページベクタデータ 2 2 0 2 内でイメージ 2 2 0 3 として構成される。このページベクタデータ 2 2 0 2 をタイルベクタデータに変換すると、図 2 2 (c) のようになる。

20

【 0 3 1 8 】

オブジェクトの分割処理 (図 6 のステップ S 6 0 5) による、イメージオブジェクトの分割処理では、まず、ステップ S 1 5 0 1 において注目タイル 2 2 0 4 を選択する (図 1 6 (c)) 。

【 0 3 1 9 】

そして、図 1 6 (d) に示すように、続いて、注目タイル 2 2 0 4 がイメージ座標へ投影される (ステップ S 1 5 0 2) 。次に、イメージ座標に投影されたタイル 2 2 0 5 のタイル境界とイメージオブジェクト 2 2 0 1 自身の外周からなる凸多角形 2 2 0 6 を算出し、この凸多角形 2 2 0 6 に外接し、かつ直立した長方形 2 2 0 7 を算出する (ステップ S 1 5 0 3) 。

30

【 0 3 2 0 】

次に、このイメージオブジェクト 2 2 0 1 の解像度と、所定解像度とを比較する (ステップ S 1 5 0 4) 。イメージオブジェクト 2 2 0 1 の解像度が所定解像度よりも大きい場合 (高解像度である場合) 、図 2 2 (e) に示すように、この凸多角形 2 2 0 6 に外接し、かつ直立した長方形 2 2 0 7 を補間処理の参照領域分拡大した拡大長方形 2 2 0 8 を算出する (ステップ S 1 5 0 7) 。

【 0 3 2 1 】

次に、拡大長方形 2 2 0 8 を境界として、図 2 2 (f) に示すように、イメージオブジェクト 2 2 0 1 から分割されたイメージオブジェクト 2 2 0 9 を生成する (ステップ S 1 5 1 0) 。その後、図 2 2 (g) に示すように、注目タイル 2 2 1 0 に埋め込まれる (ステップ S 1 5 1 1) 。注目タイル 2 2 1 0 は、タイル内座標を独自に持ち、分割されたイメージオブジェクト 2 2 0 9 は、タイル内座標に、イメージオブジェクト 2 2 0 9 として投影される。

40

【 0 3 2 2 】

次に、図 7 のステップ S 7 7 において、イメージオブジェクトをラスタライズする場合の処理の詳細について、図 2 3 を用いて説明する。

【 0 3 2 3 】

図 2 3 は本発明の実施形態 4 のステップ S 7 7 における、イメージオブジェクトのラスタライズの詳細を示すフローチャートである。

【 0 3 2 4 】

50

(ステップS1701)

まず、タイル内座標系と、デバイス座標系の対応関係を示す第1変換行列を取得する。ここで、デバイス座標系とは、タイルベクタデータから構成されるページ全体に対して定義される座標系であり、例えば、図8で示される座標系である。

【0325】

(ステップS1702)

イメージオブジェクトのイメージ座標系と、タイル内座標との対応関係を示す第2変換行列を取得する。

【0326】

(ステップS1703)

ステップS1701で取得した第1変換行列と、ステップS1702で取得した第2変換行列の2つの変換行列を合成した第3変換行列を生成し、この第3変換行列を用いて、デバイス座標系とイメージ座標系の対応関係を算出する。

【0327】

(ステップS1704)

デバイス座標系において、イメージオブジェクトが配置される領域内で、注目画素を選択する。

【0328】

(ステップS1705)

ステップS1703で合成された第3変換行列を用いて、注目画素のデバイス座標系のデバイス座標からイメージ座標系のイメージ座標の該当画素を算出する。

【0329】

(ステップS1706)

イメージオブジェクトの該当画素がタイル境界外に存在するか否かに基づいて、イメージオブジェクトの補間方法を選択する。タイル境界外に存在しない場合、補間方法として、第1補間方法(ニアレストネイバー法)を選択して、ステップS1707へ進む。一方、タイル境界外に該当画素の周辺画素が存在する場合、補間方法として、第2補間方法(バイキュービック法)を選択して、ステップS1708へ進む。

【0330】

尚、この補間方法としては、ニアレストレイバー法、バイキュービック法、バイリニア法等の各種補間方法を利用することができる。

【0331】

(ステップS1707)

該当画素の色を取得する。

【0332】

(ステップS1708)

該当画素と周辺画素を取得する。

【0333】

(ステップS1709)

該当画素と周辺画素を用いて、補間処理を実行し、処理結果の色を取得する。

【0334】

(ステップS1710)

取得した色を注目画素の色として設定する。

【0335】

(ステップS1711)

イメージオブジェクトが配置されるデバイス座標系のデバイス座標上の全ての画素を処理したか否かを判定する。全ての画素を処理していない場合(ステップS1711でNO)、ステップS1704へ戻る。一方、全ての画素を処理している場合(ステップS1711でYES)、処理を終了する。

【0336】

10

20

30

40

50

以上説明したように、実施形態 4 によれば、実施形態 1 ~ 3 で説明した効果に加えて、処理対象のイメージオブジェクトの解像度に基づいて、イメージオブジェクトのタイル分割方法を適応的に切り替えてタイルベクタデータを生成する。これにより、イメージオブジェクトのタイル分割によるデータ量増大を抑え、かつ補間処理を行わないことによるエイリアシングの発生を抑えることが可能となる。

【 0 3 3 7 】

[実施形態 5]

上述の各実施形態では、画像処理システム内のデータハンドリングをリアルタイムで行える高速化に実行できるようために、タイルベクタデータを生成して、HDD 8 で管理する構成としている。これに対し、実施形態 5 では、HDD 8 へのタイルベクタデータを格納する時に、それを構成するタイル中にオブジェクトが存在しないタイルは、HDD 8 に書き込まないで管理する。

10

【 0 3 3 8 】

これにより、HDD 8 のデータアクセスの高速化を図り、画像処理システムとしてのパフォーマンスをより向上することができる。

【 0 3 3 9 】

以下、実施形態 5 における処理概要及び処理フローについて、図 2 4 及び図 2 5 を用いて説明する。

【 0 3 4 0 】

図 2 4 は本発明の実施形態 5 の処理概要を説明するための図である。また、図 2 5 は本発明の実施形態 5 のタイルベクタデータ書込処理を示すフローチャートである。

20

【 0 3 4 1 】

(ステップ S 2 5 0 1)

実施形態 5 では、システムメモリ 5 に読み込まれた 1 ページ分の画像データ (ページベクタデータあるいはラスタデータ) を、所定サイズのブロック (タイル (矩形)) に分割してタイルベクタデータを生成する。この際、各タイルベクタデータの位置を識別するタイル ID (ブロック識別情報) を生成して、これをタイルベクタデータのヘッダに設定する。

【 0 3 4 2 】

例えば、図 2 4 (a) のページベクタデータ 2 4 0 1 を、1 辺が 8 画素四方のタイルで分割する場合、このページベクタデータ 2 4 0 1 は、X 方向に 3 個分、Y 方向に 3 個分の分割されるとする。これにより、点線で区切られた 3 × 3 合計 9 個のタイルベクタデータに分割される。尚、このタイルの大きさの単位となる 1 辺の画素数は、ユーザ操作によって変更可能である。

30

【 0 3 4 3 】

各タイルは、X 方向に左から 0、1、2、Y 方向に上から 0、1、2 とすることにより、各タイルの座標を規定できる。そして、これらの座標の組み合わせが各タイルの位置を識別するタイル ID として利用することができる。例えば、左上のタイルは、タイルベクタデータの開始となるタイルとなり、そのタイル ID は (X , Y) = (0 , 0) となる。また、その右側のタイルのタイル ID は (X , Y) = (1 , 0)、一番右下のタイルのタイル ID は (X , Y) = (3 , 3) となる。

40

【 0 3 4 4 】

(ステップ S 2 5 0 2)

システムメモリ 5 に読み込まれた、1 ページ分のページベクタデータに存在するオブジェクトを、タイル内に収まる複数のオブジェクトに分割して、タイルベクタデータを生成する。そのために、タイル内のオブジェクトの有無を識別する識別フラグ (オブジェクト識別情報) を生成し、これをタイルベクタデータのヘッダに設定する。

【 0 3 4 5 】

具体的には、ページベクタデータ 2 4 0 1 を構成する各タイルにおいて、オブジェクトの有無を解析する。オブジェクトがある場合は、識別フラグを「1」、オブジェクトがな

50

い場合は「0」に設定する。

【0346】

例えば、タイルベクタデータ2401中のタイルID=(0,0)、(1,0)、(2,0)、(0,1)、(0,2)のタイル内には、オブジェクトが存在しないため、識別フラグは「0」となる。一方、残りのタイルID=(1,1)、(1,2)、(2,1)、(3,3)のタイル内には、オブジェクトが存在するため、識別フラグは「1」となる。

【0347】

(ステップS2503)

各タイルの位置を識別するタイルID、タイル内のオブジェクトの有無を示す識別フラグ及び各タイルベクタからタイルベクタデータを生成する。そのために、システムメモリ5に、図24(b)に示すタイルテーブル2402(タイル管理テーブル)を作成する。このタイルテーブル2402では、タイルID、識別フラグ、オブジェクトの内容(オブジェクトなし(n o o b j e c t)、データ(D a t a)等)の情報を、タイル毎に管理する。この場合、タイル名T i l e 0 ~ T i l e 8のタイルからなるタイルテーブルが構成される。

【0348】

(ステップS2504)

システムメモリ5に読み込まれた、1ページ分のページベクタデータ2401に対するタイルテーブル2402の生成が完了すると、システムメモリ5からHDD8へのタイルベクタデータの書込を開始するための書込設定を実行する。ここの書込設定においては、タイル位置を識別するタイルID=(0,0)のタイルデータから書込を開始するための設定を実行する。

【0349】

(ステップS2505)

タイルIDに基づいて、処理対象のタイルのタイルテーブル2402の識別フラグを参照する。そして、その識別フラグ=1であるか否かを判定する。識別フラグ=1である場合(ステップS2505でYES)、タイル内にオブジェクトが存在するので、このタイルは、HDD8への書込対象であると判定して、ステップS2506へ進む。一方、識別フラグ=0である場合(ステップS2505でNO)、タイル内にオブジェクトが存在しないので、このタイルは、HDD8への書込対象でないと判定して、ステップS2507へ進む。

【0350】

(ステップS2506):

処理対象のタイルのHDD8への書込を実行する。

【0351】

(ステップS2507):

未処理タイルデータの有無を判定する。未処理タイルデータがある場合(ステップS2507でYES)、ステップS2504へ戻り、次のタイルIDが示すタイルデータを処理対象のタイルデータとして設定する。一方、未処理タイルデータがない場合(ステップS2507でNO)、処理を終了する。

【0352】

ここで、ステップS1504~ステップS1507の処理の具体的な動作について、図24(a)を用いて説明する。

【0353】

まず、ステップS2504で、タイルID=(0,0)を設定する。これによって、タイルID=(0,0)の識別フラグをタイルテーブル2402で参照すると、その値は「0」であることから、対応するタイル内にオブジェクトは存在しないと判定する。そのため、このタイルID=(0,0)のタイルデータ(T i l e 0)は、HDD8への書込は実行しない(書込を禁止する)。

10

20

30

40

50

【 0 3 5 4 】

従って、この場合のタイルID = (0 , 0) のタイルについての処理は、ステップS 2 5 0 5 からステップS 2 5 0 7 へ進む。ステップS 2 5 0 7 では、未処理のタイル (タイルID) が存在するので、ステップS 2 5 0 4 へ戻る。

【 0 3 5 5 】

次に、ステップS 2 5 0 4 で、タイルID = (1 , 0) を設定する。これによって、タイルID = (1 , 0) の識別フラグをタイルテーブル2 4 0 2 で参照すると、その値は「 0 」であることから、対応するタイル内にオブジェクトは存在しないと判定する。そのため、このタイルID = (1 , 0) のタイルデータ (T i l e 1) は、H D D 8 への書込は実行しない (書込を禁止する) 。

10

【 0 3 5 6 】

同様にして、タイルID = (2 , 0) 、 (0 , 1) のタイルデータ (T i l e 2 、 T i l e 3) も識別フラグの値が「 0 」であることから、そのタイルデータのH D D 8 への書込動作を禁止する。

【 0 3 5 7 】

次のタイルID = (1 , 1) の識別フラグをタイルテーブル2 4 0 2 で参照すると、その値は「 1 」であることから、対応するタイル内にオブジェクトは存在すると判定する。そのため、このタイルID = (1 , 1) のタイルデータ (T i l e 4) は、H D D 8 への書込を実行する。

【 0 3 5 8 】

20

従って、この場合のタイルID = (1 , 1) のタイルについての処理は、ステップS 2 5 0 5 からステップS 2 5 0 6 へ進む。そして、書込が終了すると、未処理のタイル (タイルID) が存在するので、ステップS 2 5 0 4 へ戻る。

【 0 3 5 9 】

以下、同様にして、タイルID = (2 , 1) のタイルデータ (T i l e 5) も識別フラグの値が「 1 」であることから、そのタイルデータのH D D 8 への書込動作を実行する。次のタイルID = (0 , 2) のタイルデータ (T i l e 6) は識別フラグの値が「 0 」であることから、そのタイルデータのH D D 8 への書込動作を禁止する。更に、タイルID = (1 , 2) 、 (2 , 2) のタイルデータ (T i l e 7 、 T i l e 8) の識別フラグの値が「 1 」であることから、そのタイルデータのH D D 8 への書込動作を実行する。

30

【 0 3 6 0 】

そして、タイルID = (2 , 2) に対する処理が完了した時点で、未処理のタイルがなくなるので、書込処理を終了する。

【 0 3 6 1 】

以上の処理によって、タイルテーブル2 4 0 2 で管理されるタイル名T i l e 0 ~ T i l e 8 の内、H D D 8 には、タイル名T i l e 4 、 5 、 7 及び8 のタイルデータだけが格納されることになるので、H D D 8 の記憶容量を節約することができる。また、H D D 8 のタイルデータの格納状態として、タイルテーブル2 4 0 4 が生成される。

【 0 3 6 2 】

次に、H D D 8 に書き込まれたタイルベクタデータをシステムメモリ5 へ読み出す場合の読出処理について、図2 6 を用いて説明する。

40

【 0 3 6 3 】

図2 6 は本発明の実施形態5 のタイルベクタデータ読出処理を示すフローチャートである。

【 0 3 6 4 】

(ステップS 2 6 0 1)

H D D 8 に存在するタイルデータを読み出して識別するためのタイルID の読出設定を実行する。

【 0 3 6 5 】

(ステップS 2 6 0 2)

50

HDD 8 に格納されているタイルテーブル 2404 を参照して、HDD 8 に実際に格納されているタイルデータのタイル ID を読み出してシステムメモリ 5 に記憶する。

【0366】

(ステップ S2603)

読出設定で設定した設定タイル ID と、HDD 8 から読み出した読出タイル ID が一致しているか否かを判定することで、読出順序が不連続のタイル ID の有無を判定する。一致していない場合 (ステップ S2603 で YES)、不連続タイル ID があると判定し、ステップ S2604 へ進む。一方、一致している場合 (ステップ S2603 で NO)、不連続タイル ID がないと判定し、ステップ S2605 へ進む。

【0367】

(ステップ S2604)

不連続タイル ID がある場合は、内部にオブジェクトが存在しないために、HDD 8 へ書き込まれなかったタイルデータが存在することになる。この場合、読出時にはそのタイルデータを、それを構成要素とするタイルベクタデータを再現するために、タイルテーブル 2404 の状態から、タイルテーブル 2402 の状態へ再構成する必要がある。

【0368】

そこで、ここでは、タイルテーブル 2404 に対して、不連続タイル ID に該当するタイルデータに関するデータを、タイルテーブル 2404 に追加する。

【0369】

(ステップ S2605)

不連続タイル ID がない場合は、そのまま、タイルテーブル 2404 を参照して、タイルデータの読出を実行する。

【0370】

(ステップ S2606) :

タイルデータの読出の実行後、未処理のタイルの有無を判定する。未処理タイルデータがある場合 (ステップ S2606 で NO)、ステップ S2601 へ戻り、次のタイル ID が示すタイルデータを処理対象のタイルデータとして設定する。一方、未処理タイルデータがない場合 (ステップ S2606 で YES)、処理を終了する。

【0371】

ここで、ステップ S2601 ~ ステップ S2606 の処理の具体的な動作について、図 24(a) を用いて説明する。

【0372】

まず、ステップ S2601 で、HDD 8 からタイルデータの読出を実行するために、その読出位置とするタイル位置を先頭 ID = (0, 0) (設定タイル ID) に設定する。

【0373】

ステップ S2602 で、タイルテーブル 2404 を参照して、HDD 8 からタイル ID の読出を実行し、読み出したタイル ID (読出タイル ID) をシステムメモリ 5 に記憶する。読み出したタイル ID は、後のタイルテーブルへのタイルデータの追加処理で使用するため、最新の読出タイル ID 履歴をいくつか保存しておく。ここでは、HDD 8 に記憶されているタイルテーブル 2404 からタイル ID = (1, 1) を読み出す。

【0374】

ステップ S2603 で、タイル ID の比較を行う。ここでは、設定タイル ID = (0, 0) に対し、読出タイル ID = (1, 1) であるため、タイル ID の読出順序が不連続、つまり、不連続タイル ID となっていることが分かる。不連続タイル ID があるということは、内部にオブジェクトが存在しないタイルデータが存在することを示している。

【0375】

そのため、ここでは、ステップ S2604 で、タイルテーブルへのタイルデータの追加を実行する。つまり、読出タイル ID = (1, 1) のタイルデータにはオブジェクトデータが存在するため、それより前の不連続分のタイル ID = (0, 0)、(1, 0)、(2, 0)、(0, 1) に関して、オブジェクトが存在しないタイルデータを追加したタイル

10

20

30

40

50

テーブル 2 4 0 2 を作成する。この時点では、タイルテーブル 2 4 0 2 は、タイル名が T i l e 0 ~ T i l e 3 までが構成される。

【 0 3 7 6 】

次に、ステップ S 2 6 0 5 で、タイル I D = (1 , 1) のタイルデータを H D D 8 から読み出す。ステップ S 2 6 0 6 では、未処理のタイル I D が存在するので、ステップ S 2 6 0 1 へ戻る。

【 0 3 7 7 】

次に、ステップ S 2 6 0 1 では、設定タイル I D = (2 , 1)、ステップ S 2 6 0 2 では、読出タイル I D = (2 , 1) となる。また、ステップ S 2 6 0 2 では、読出タイル I D の履歴をシステムメモリ 5 に保存してあるため、ステップ S 2 6 0 3 で、設定タイル I D = (2 , 1) に対して、前回の読出タイル I D = (1 , 1) との比較を行う。今回は、タイル I D の読出順序が連続しているため、ステップ S 2 6 0 5 へ進み、タイル I D = (2 , 1) のタイルデータを H D D 8 から読み出す。

【 0 3 7 8 】

以下、同様にして、処理を実行することにより、タイル I D = (0 , 2) について、そのタイルデータがタイルテーブル 2 4 0 2 に追加され、タイル I D = (1 , 2)、(2 , 2) については、H D D 8 からのタイルデータの読出が実行される。

【 0 3 7 9 】

これにより、未処理のタイル I D がなくなるので、読出処理を終了する。この読出処理が終了することで、書込処理時と同じタイルテーブル 2 4 0 2 が再現される。そして、このタイルテーブル 2 4 0 2 を用いることで、タイルベクタデータ 2 4 0 5 を復元（読出）することができる。読出処理の終了後、読出タイル I D の履歴は消去される。

【 0 3 8 0 】

以上説明したように、実施形態 5 によれば、実施形態 1 ~ 4 で説明した効果に加えて、システムメモリ上に生成されたタイルベクタデータを H D D へ格納する場合には、そのタイルベクタデータを構成する各タイルデータの位置を識別するタイルヘッダ及び、オブジェクトの有無を識別するフラグを管理するタイルテーブルを作成し、オブジェクトが存在しないタイルデータに関しては、H D D に該当するタイルデータの書込を禁止する。

【 0 3 8 1 】

これにより、H D D へ格納するデータ量を削減することが可能となるとともに、H D D への書込 / 読出処理において、アクセス回数を削減でき、動作パフォーマンスの高いシステムが構成を実現することができる。

【 0 3 8 2 】

[実施形態 6]

図 2 9 に示す従来のシステムでは、ページベクタデータに含まれるフォントオブジェクトは、C P U 1 0 2 によりシステムメモリ 1 0 4 上に設けられたフォントキャッシュ（不図示）に、対応するフォントオブジェクトが既に登録されているか否かが検索される。そして、対応するフォントオブジェクトが登録されていない場合、C P U 1 0 2 により、そのフォントオブジェクトがビットマップ展開された後、フォントキャッシュに登録される。

【 0 3 8 3 】

そして、その登録されたフォントデータのシステムメモリ 1 0 4 上の位置が中間データ内で指示され登録されていれば、展開処理無しに、その展開済フォントデータのシステムメモリ 1 0 4 上の位置が中間データ内で指示される。

【 0 3 8 4 】

中間データを受け取った画像データ展開部 1 1 3 は、フォント以外の描画オブジェクトを展開処理すると共に、指示されたフォントキャッシュの位置より展開済フォントデータを読み取り、ビットマップデータを作成して、システムメモリ 1 0 4 へ保存する。その後、S B B 1 0 1 を介して画像処理部 1 1 0 で画像処理され、プリンタ 1 1 2 へ送られ印刷される。

10

20

30

40

50

【0385】

このような画像処理システムにおいて、フォントキャッシュの有効活用を行なう構成として、例えば、特開平10-016319号公報の構成がある。この構成では、フォントキャッシュデータ毎にカウンタを設け、中間データ生成時にフォントキャッシュにヒットする毎に、カウンタをインクリメントする。また、画像データ展開処理にてヒットしたフォントキャッシュを参照する毎にカウンタをデクリメントする。そして、カウンタが初期値になるまで、フォントキャッシュデータを保持する。

【0386】

しかしながら、このような構成では、ヒットする回数分以上のカウンタを用意する必要がある。ここで、通常、処理対象のデータ内に一つの文字が何回出現するかは予想できないので十分大きなカウンタが必要になる。通常、カウンタはフォントキャッシュと同じシステムメモリ上に用意されることになるので、より大きな容量のシステムメモリを必要としてしまう。

10

【0387】

また、さらなる印刷速度向上を狙って、インタプリタ部と画像データ展開部を複数具備し、並列処理を実現した際には、カウンタに対して複数のインタプリタ部と複数の画像データ展開部からのアクセスが同時に発生してしまうことが多発し、並列処理を阻害する可能性も大きくなる。

【0388】

そこで、実施形態6では、画像データ展開部が備える複数の小画像データ展開部が並列にラスタライズ処理（ベクタデータをビットマップデータへ変換）を行なう場合に、各小画像データ展開部が参照しているフォントキャッシュデータを確実に保護する構成について説明する。

20

【0389】

このフォントキャッシュを保護するための構成は、各小画像データ展開部が同時に（独立に）操作できるようにすることで、画像データ展開処理の並列動作を阻害することを防止することができる。

【0390】

また、このフォントキャッシュを保護するための構成は、該当フォントの出現回数には因らず、並列に動作する小画像データ展開部の個数分だけ用意すればよいので、メモリの使用量を削減することができる。

30

【0391】

以下、実施形態6における、画像データ展開部18の応用例について説明する。

【0392】

まず、画像データ展開部18の詳細構成について、図27を用いて説明する。

【0393】

図27は本発明の実施形態6の画像データ展開部の詳細構成を示す図である。

【0394】

小画像データ展開部（ μ RIP）18a～18dには、それぞれが並列に入力されるタイルベクタを展開することでパフォーマンス向上を果たしている。2705は、一度展開した文字（フォント）データを再利用のために保存するフォントキャッシュである。フォントキャッシュ2705には、フォントキャッシュメモリ2706と、フォントキャッシュメモリ2706を参照中であることを示すロックフラグを記憶するロックフラグレジスタ2707～2710より構成されている。尚、図27では図示していないが、フォントキャッシュメモリ2706は、図1のローカルメモリ19上に割り当てられる。

40

【0395】

2711は各 μ RIP18a～18dが共通のフォントキャッシュメモリ2706へアクセスするためのバスであり、各 μ RIP18a～18dが展開すべき文字がフォントキャッシュ2705に登録されていれば、展開済のフォントデータを2711を介してリードするだけで文字の展開が終了する。

50

【0396】

2712～2715は、各μRIP18a～18dより独立に対応するロックフラグレジスタ2707から2710を操作するための制御信号線である。μRIP18aはロックフラグレジスタ2707へ、μRIP18bはロックフラグレジスタ2708へ、μRIP18cはロックフラグレジスタ2709へ、μRIP18dはロックフラグレジスタ2710へ接続されている。

【0397】

2716～2719はそれぞれのμRIP18a～18dで展開処理対象のタイルベクタデータの一例を示している。

【0398】

尚、図27において、μRIPとロックフラグレジスタは4組示されているが、この数に限定されるものではなく、要求されるシステムのパフォーマンスに合わせて増減して良いことは言うまでもない。

【0399】

次に、フォントキャッシュの動作例について、図28を用いて説明する。

【0400】

図28は本発明の実施形態6のフォントキャッシュの動作例を示す図である。

【0401】

図28は、フォントキャッシュメモリ2706とロックフラグレジスタ2707～2710で構成されるフォントキャッシュ2705の一例を示している。

【0402】

2100は、フォントキャッシュ2705のデータを管理するためのフォントキャッシュ管理テーブルである。2101は、n個の格納部から構成される展開済フォントデータの格納部である。フォントキャッシュ管理テーブル2100には登録されている文字を識別するための文字識別情報と、その文字の展開済フォントデータが格納されているフォントキャッシュメモリのアドレス（格納アドレス）、及び、小画像データ展開部（μRIP）の数だけのロックフラグレジスタで構成されている。

【0403】

展開済フォントデータの格納部2101には、フォントキャッシュ管理テーブル2100に示される格納アドレスから、各文字の展開済フォントデータが格納されている。

【0404】

以下、図27及び28を用いて、各μRIPのフォント展開処理とフォントキャッシュの動作について説明する。

【0405】

μRIP18aは、タイルベクタデータ2716を受け取る。このタイルベクタデータ2716には、文字「あ」が含まれている。μRIP18aは、フォントキャッシュ管理テーブル2100の文字識別情報を検索し、テーブル内のデータエントリ2103で「あ」が登録済であることを認識する（ヒットと認識する）。

【0406】

この場合、ヒットしたので、μRIP18aは、制御信号線2712によって接続されているロックフラグレジスタ2707のデータエントリ2103に相当するフラグに「1」をセットし、フォントキャッシュメモリ2706を参照中であることを示す。

【0407】

その後、フォントキャッシュ管理テーブル2100内のデータエントリ2103の格納アドレスで示される「Address 1」より、展開済フォントデータ1（2109）の読出を開始し、タイルベクタデータ2716の展開処理を実行する。展開済フォントデータ1（2109）の読出が終了したら、先にセットしたロックフラグレジスタ2707のデータエントリ2103に相当するフラグを「0」にクリアし、フォントキャッシュメモリ2706の参照が終了したことを示す。

【0408】

10

20

30

40

50

μ R I P 1 8 b は、タイルベクタデータ 2 7 1 7 を受け取る。このタイルベクタデータ 2 7 1 7 には文字「A」が含まれている。μ R I P 1 8 b は、フォントキャッシュ管理テーブル 2 1 0 0 の文字識別情報を検索し、テーブル内の 2 データエントリ 2 1 0 5 で「A」が登録済であることを認識する（ヒットと認識する）。

【 0 4 0 9 】

この場合、ヒットしたので、μ R I P 1 8 b は、制御信号線 2 7 1 3 によって接続されているロックフラグレジスタ 2 7 0 8 のデータエントリ 2 1 0 5 に相当するフラグに「1」をセットし、フォントキャッシュメモリ 2 7 0 6 を参照中であることを示す。

【 0 4 1 0 】

その後、フォントキャッシュ管理テーブル 2 1 0 0 内のデータエントリ 2 1 0 5 の格納アドレスで示される「A d d r e s s 3」より、展開済フォントデータ 3（2 1 1 1）の読出を開始し、タイルベクタデータ 2 7 1 7 の展開処理を実行する。展開済フォントデータ 3（2 1 1 1）の読出が終了したら、先にセットしたロックフラグレジスタ 2 7 0 8 のデータエントリ 2 1 0 5 に相当するフラグを「0」にクリアし、フォントキャッシュメモリ 2 7 0 6 の参照が終了したことを示す。

【 0 4 1 1 】

μ R I P 1 8 c は、タイルベクタデータ 2 7 1 8 を受け取る。このタイルベクタデータ 2 7 1 8 には文字「A」が含まれている。μ R I P 1 8 c は、フォントキャッシュ管理テーブル 2 1 0 0 の文字識別情報を検索し、テーブル内のデータエントリ 2 1 0 5 で「A」が登録済であることを認識する（ヒットと認識する）。

【 0 4 1 2 】

この場合、ヒットしたので、μ R I P 1 8 c は、制御信号線 2 7 1 4 によって接続されているロックフラグレジスタ 2 7 0 9 のデータエントリ 2 1 0 5 に相当するフラグに「1」をセットし、フォントキャッシュメモリ 2 7 0 6 を参照中であることを示す。

【 0 4 1 3 】

その後、フォントキャッシュ管理テーブル 2 1 0 0 内のデータエントリ 2 1 0 5 の格納アドレスに示される「A d d r e s s 3」より、展開済フォントデータ 3（2 1 1 1）の読出を開始し、タイルベクタデータ 2 7 1 8 の展開処理を実行する。展開済フォントデータ 3（2 1 1 1）の読出が終了したら、先にセットしたロックフラグレジスタ 2 7 0 9 のデータエントリ 2 1 0 5 に相当するフラグを「0」にクリアし、フォントキャッシュメモリ 2 7 0 6 の参照が終了したことを示す。

【 0 4 1 4 】

μ R I P 1 8 d は、タイルベクタデータ 2 7 1 9 を受け取る。このタイルベクタデータ 2 7 1 9 には文字「う」が含まれている。μ R I P 1 8 d は、フォントキャッシュ管理テーブル 2 1 0 0 の文字識別情報を検索し、「う」が未登録であることを認識する（ミスと認識する）。

【 0 4 1 5 】

この場合、ミスしたので、μ R I P 1 8 d は文字「う」の展開処理を実行する。展開処理が終了すると、文字「う」の展開済フォントデータをフォントキャッシュメモリ 2 7 0 5 に登録する。このとき、μ R I P 1 8 d は、フォントキャッシュ管理テーブル 2 1 0 0 のロックフラグレジスタを参照し、どのフラグもセットされていないもの、つまり、他のどの μ R I P も参照中でないものの中から置き換える展開済フォントデータを選択するように動作する。ロックフラグレジスタの内容が、図 2 8 のような状態であれば、データエントリ 2 1 0 3 と 2 1 0 5 は置き換え候補から除かれることになる。

【 0 4 1 6 】

以上説明したフォントキャッシュの構成と動作により、複数の小画像データ展開部（μ R I P）が並列にラスタライズ処理（ベクタデータをビットマップデータへ変換）を行なう場合に、各小画像データ展開部が参照している展開済フォントデータ（フォントキャッシュデータ）は、ロックフラグがセットされているので置き換え対象とはならないので確実に保護できる効果がある。

10

20

30

40

50

【 0 4 1 7 】

また、ロックフラグレジスタは、各 μ RIP 毎に用意し、独立して操作できる構成にしたので、 μ RIP の並列動作を阻害する要因から排除される。さらに、ロックフラグレジスタは、 μ RIP の個数分だけ用意すればよいので、出現回数等によるカウンタによる置き換え保護と比較して、少ないリソース（メモリ等）で実現可能である。

【 0 4 1 8 】

また、実施形態 6 では、フォントキャッシュを画像データ展開部内のローカルメモリ 19 上に配置したが、システムメモリ 8 上に構成しても良いことは言うまでもない。

【 0 4 1 9 】

以上説明したように、実施形態 6 によれば、画像データ展開部が備える複数の小画像データ展開部が並列にラスタライズ処理（ベクタデータをビットマップデータへ変換）を行なう場合に、各小画像データ展開部が参照しているフォントキャッシュデータを確実に保護することができる。

10

【 0 4 2 0 】

この構成によれば、各小画像データ展開部が同時に（独立に）操作できるようにすることで、画像データ展開処理の並列動作を阻害することを防止することができる。また、該当フォントの出現回数には因らず、並列に動作する小画像データ展開部の個数分だけ用意すればよいので、メモリの使用量を削減することができる。

【 0 4 2 1 】

以上、実施形態例を詳述したが、本発明は、例えば、システム、装置、方法、プログラムもしくは記憶媒体等としての実施態様をとることが可能であり、具体的には、複数の機器から構成されるシステムに適用しても良いし、また、一つの機器からなる装置に適用しても良い。

20

【 0 4 2 2 】

尚、本発明は、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（実施形態では図に示すフローチャートに対応したプログラム）を、システムあるいは装置に直接あるいは遠隔から供給し、そのシステムあるいは装置のコンピュータが該供給されたプログラムコードを読み出して実行することによっても達成される場合を含む。

【 0 4 2 3 】

従って、本発明の機能処理をコンピュータで実現するために、該コンピュータにインストールされるプログラムコード自体も本発明を実現するものである。つまり、本発明は、本発明の機能処理を実現するためのコンピュータプログラム自体も含まれる。

30

【 0 4 2 4 】

その場合、プログラムの機能を有していれば、オブジェクトコード、インタプリタにより実行されるプログラム、OS に供給するスクリプトデータ等の形態であっても良い。

【 0 4 2 5 】

プログラムを供給するための記録媒体としては、例えば、フロッピー（登録商標）ディスク、ハードディスク、光ディスク、光磁気ディスク、MO、CD-ROM、CD-R、CD-RW、磁気テープ、不揮発性のメモリカード、ROM、DVD（DVD-ROM、DVD-R）などがある。

40

【 0 4 2 6 】

その他、プログラムの供給方法としては、クライアントコンピュータのブラウザを用いてインターネットのホームページに接続し、該ホームページから本発明のコンピュータプログラムそのもの、もしくは圧縮され自動インストール機能を含むファイルをハードディスク等の記録媒体にダウンロードすることによっても供給できる。また、本発明のプログラムを構成するプログラムコードを複数のファイルに分割し、それぞれのファイルを異なるホームページからダウンロードすることによっても実現可能である。つまり、本発明の機能処理をコンピュータで実現するためのプログラムファイルを複数のユーザに対してダウンロードさせる WWW サーバも、本発明に含まれるものである。

【 0 4 2 7 】

50

また、本発明のプログラムを暗号化してＣＤ－ＲＯＭ等の記憶媒体に格納してユーザに配布し、所定の条件をクリアしたユーザに対し、インターネットを介してホームページから暗号化を解く鍵情報をダウンロードさせ、その鍵情報を使用することにより暗号化されたプログラムを実行してコンピュータにインストールさせて実現することも可能である。

【０４２８】

また、コンピュータが、読み出したプログラムを実行することによって、前述した実施形態の機能が実現される他、そのプログラムの指示に基づき、コンピュータ上で稼動しているＯＳなどが、実際の処理の一部または全部を行ない、その処理によっても前述した実施形態の機能が実現され得る。

【０４２９】

10

さらに、記録媒体から読み出されたプログラムが、コンピュータに挿入された機能拡張ボ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き込まれた後、そのプログラムの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるＣＰＵなどが実際の処理の一部または全部を行ない、その処理によっても前述した実施形態の機能が実現される。

【図面の簡単な説明】

【０４３０】

【図１】本発明の実施形態１の画像処理システムを構成するＭＦＰのコントローラの詳細を示すブロック図である。

【図２】本発明の実施形態１の画像処理システムでのコピー動作に係るデータフローを示す図である。

20

【図３】本発明の実施形態１の画像処理システムでのプリント動作に係るデータフローを示す図である。

【図４】本発明の実施形態１の画像処理システムでの送信動作に係るデータフローを示す図である。

【図５】本発明の実施形態１のラスタ／タイルベクタ変換部が実行する処理を示すフローチャートである。

【図６】本発明の実施形態１のネットワークから転送されてくるドキュメントデータの一例を示している。

【図７】本発明の実施形態１のページベクタデータの記述例を示す図である。

30

【図８】本発明の実施形態１のタイルベクタデータの例を示す図である。

【図９】本発明の実施形態１のタイルベクタデータの記述例を示す図である。

【図１０】本発明の実施形態１のタイル／ページベクタ変換部が実行する処理を示すフローチャートである。

【図１１】本発明の実施形態１の画像データ展開部が実行する処理を示すフローチャートである。

【図１２】本発明の実施形態２の画像処理システムでの送信動作に係るデータフローを示す図である。

【図１３】本発明の実施形態３のタイルベクタデータの例を示す図である。

【図１４】本発明の実施形態３のページベクタデータの記述例を示す図である。

40

【図１５】本発明の実施形態３のタイルベクタデータの記述例を示す図である。

【図１６】本発明の実施形態３の曲線オブジェクトのタイル分割を説明するための図である。

【図１７】本発明の実施形態３の曲線オブジェクトの分割処理の詳細を示すフローチャートである。

【図１８】本発明の実施形態３のステップＳ１８０２の処理の詳細を示すフローチャートである。

【図１９】本発明の実施形態３のステップＳ１８０３の処理の詳細を示すフローチャートである。

【図２０】本発明の実施形態４のイメージオブジェクトのタイル分割を説明するための図

50

である。

【図 2 1】本発明の実施形態 4 のイメージオブジェクトの分割処理の詳細を示すフローチャートである。

【図 2 2】本発明の実施形態 4 のイメージオブジェクトの分割処理の具体例を示す図である。

【図 2 3】本発明の実施形態 4 のステップ S 7 7 における、イメージオブジェクトのラスタライズの詳細を示すフローチャートである。

【図 2 4】本発明の実施形態 5 の処理概要を説明するための図である。

【図 2 5】本発明の実施形態 5 のタイルベクタデータ書込処理を示すフローチャートである。

10

【図 2 6】本発明の実施形態 5 のタイルベクタデータ読出処理を示すフローチャートである。

【図 2 7】本発明の実施形態 6 の画像データ展開部の詳細構成を示す図である。

【図 2 8】本発明の実施形態 6 のフォントキャッシュの動作例を示す図である。

【図 2 9】従来の画像処理システムの構成を示す図である。

【符号の説明】

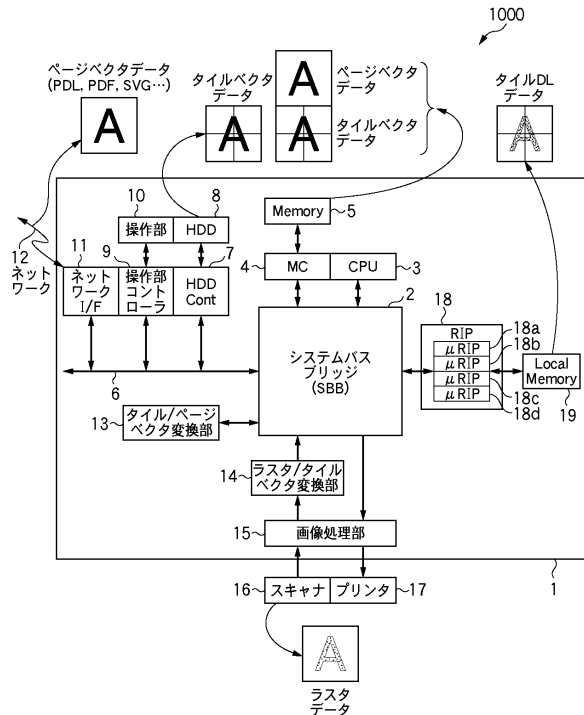
【 0 4 3 1 】

- 1 コントローラ
- 2 システムバスブリッジ
- 3 C P U
- 4 メモリコントローラ
- 5 システムメモリ
- 6 汎用バス
- 7 ハードディスクコントローラ
- 8 ハードディスクドライブ
- 9 操作部コントローラ
- 10 操作部
- 11 ネットワーク I / F
- 12 ネットワーク
- 13 タイル / ページベクタ変換部
- 14 ラスタ / タイルベクタ変換部
- 15 画像処理部
- 16 スキャナ
- 17 プリンタ
- 18 画像データ展開部
- 18 a ~ 18 d 小画像データ展開部
- 19 ローカルメモリ

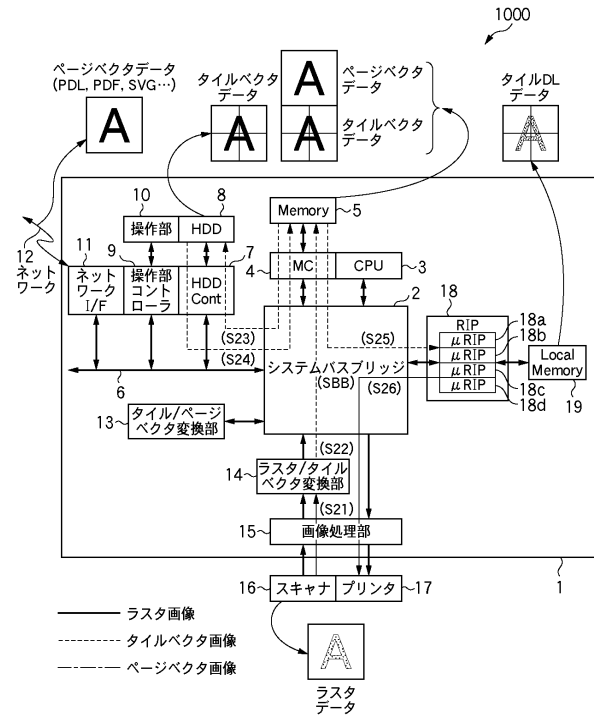
20

30

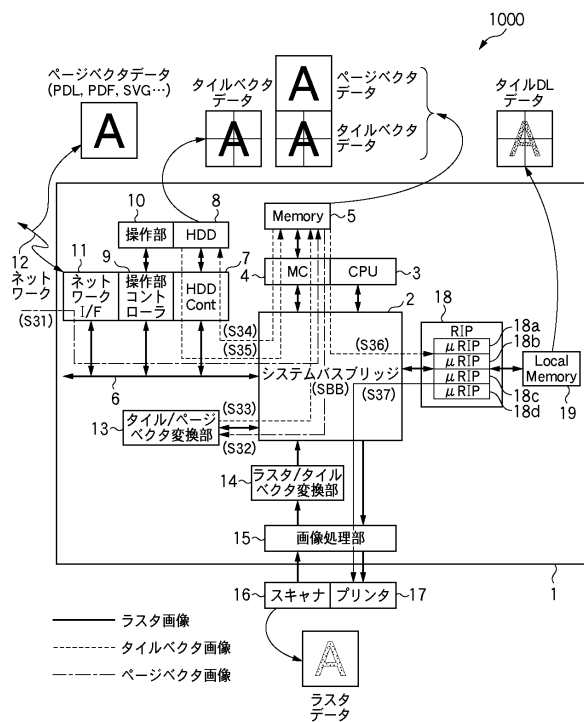
【図 1】



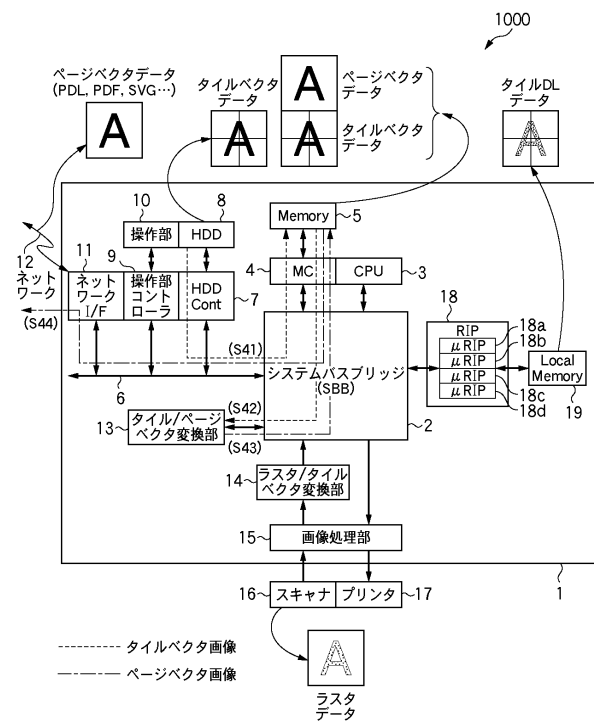
【図 2】



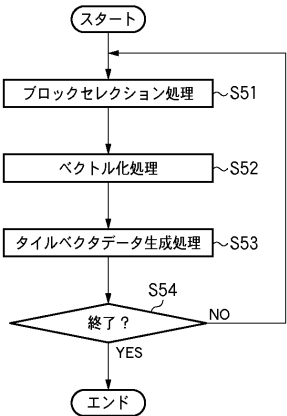
【図 3】



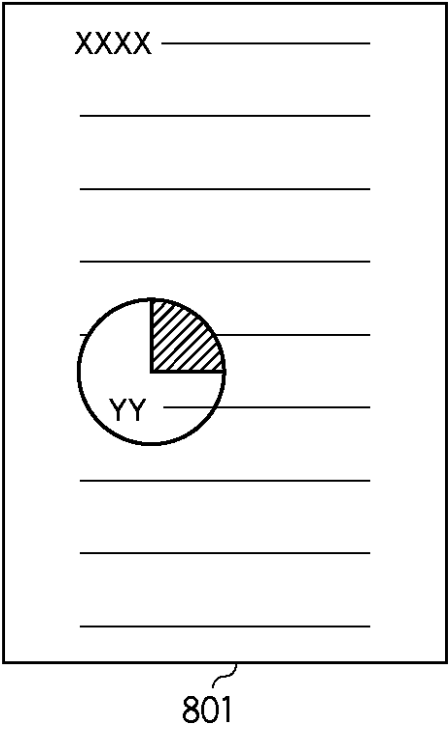
【図 4】



【図 5】



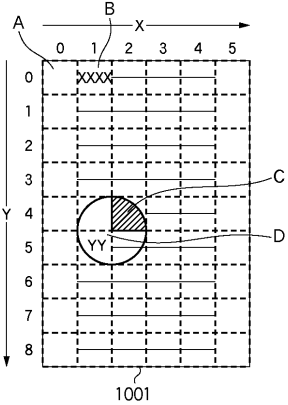
【図 6】



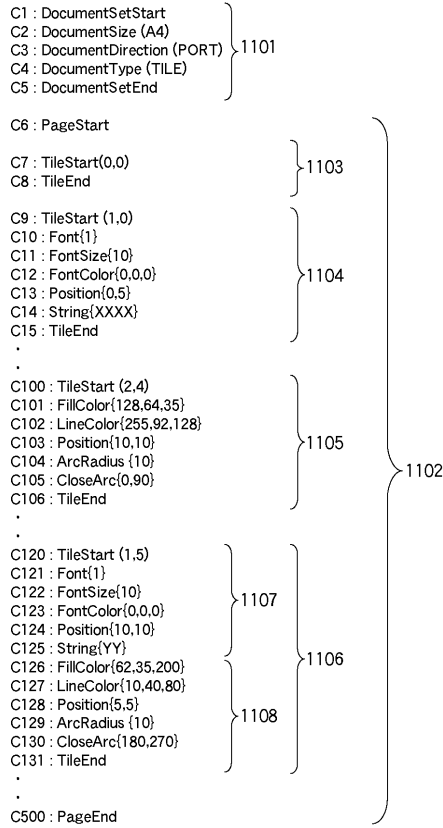
【図 7】

- C1 : DocumentSetStart
- C2 : DocumentSize (A4)
- C3 : DocumentDirection (PORT)
- C4 : DocumentType (Page)
- C5 : DocumentSetEnd
- } 901
- C6 : PageStart
- C7 : Font{1}
- C8 : FontSize{10}
- C9 : FontColor{0,0,0}
- C10 : Position{10,5}
- C11 : String{XXXX...YY...}
- } 902
- C12 : FillColor{128,64,35}
- C13 : LineColor{255,92,128}
- C14 : Position{40,300}
- C15 : ArcRadius {10}
- C16 : CloseArc{0,90}
- C17 : FillColor{62,35,200}
- C18 : LineColor{10,40,80}
- C19 : Position{40,300}
- C20 : ArcRadius {10}
- C21 : CloseArc{90,360}
- C22 : PageEnd
- } 903

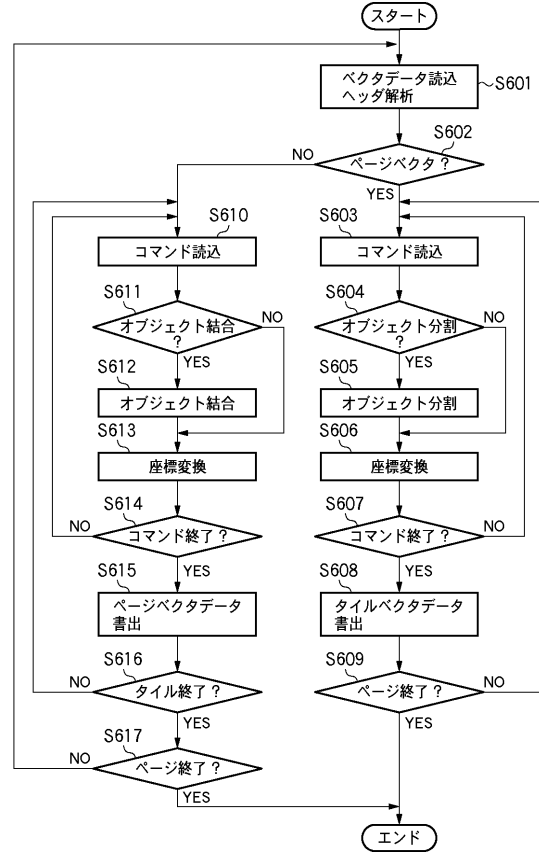
【図 8】



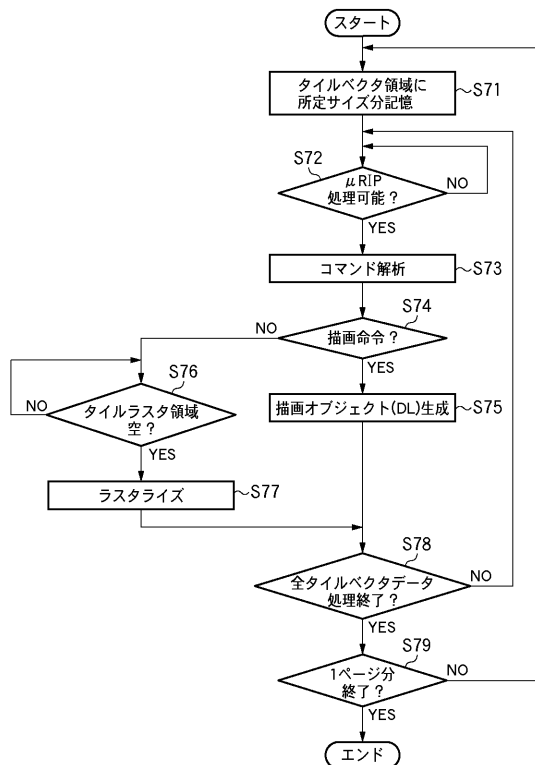
【図 9】



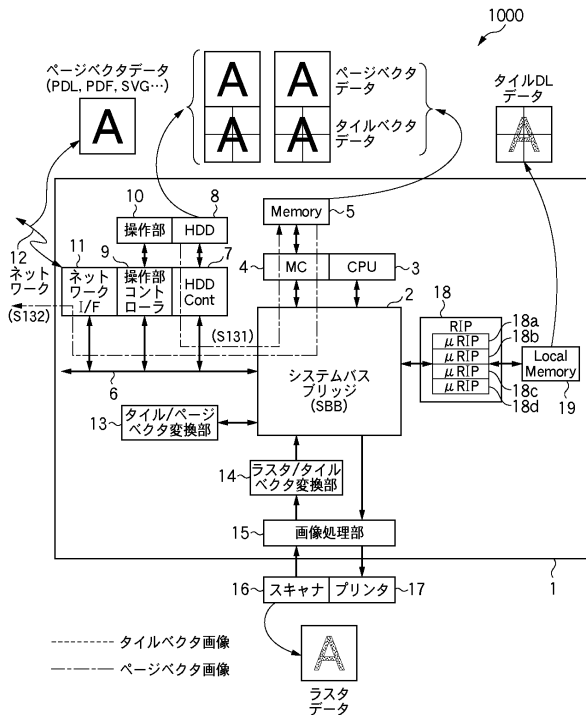
【図 10】



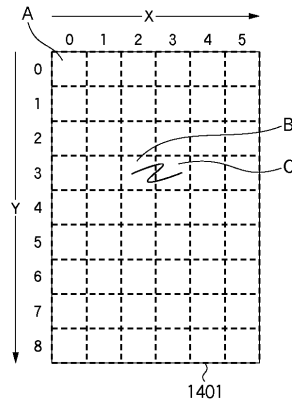
【図 11】



【図 12】



【図 13】



【図 14】

C1 : DocumentSetStart
 C2 : DocumentSize (A4)
 C3 : DocumentDirection (PORT)
 C4 : DocumentType (Page)
 C5 : DocumentSetEnd
 } 1501

 C6 : PageStart
 C7 : LineColor{255,92,128}
 C8 : Position{66,112}
 C9 : Curve{{126,98},{66,126},{126,112}}
 C10 : PageEnd
 } 1502

【図 15】

C1 : DocumentSetStart
 C2 : DocumentSize (A4)
 C3 : DocumentDirection (PORT)
 C4 : DocumentType (TILE)
 C5 : DocumentSetEnd
 } 1601

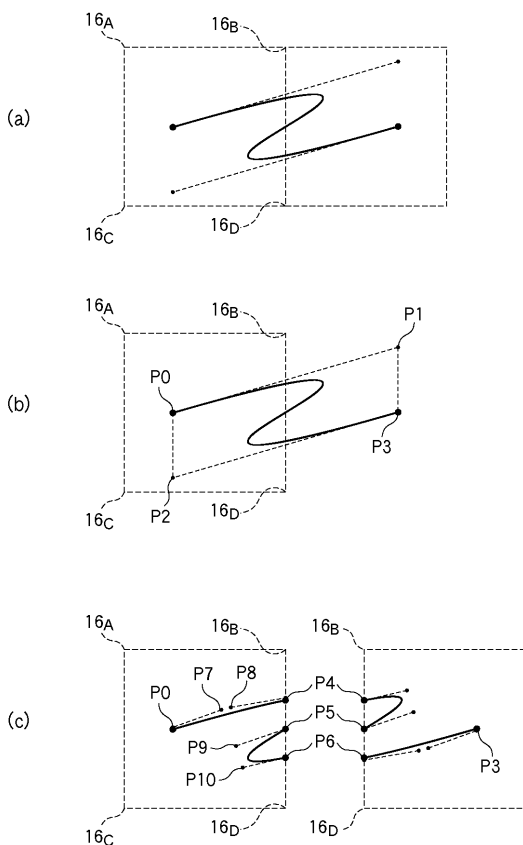
 C6 : PageStart
 C7 : TileStart(0,0)
 C8 : TileEnd
 } 1603

 C100 : TileStart (2,3)
 C101 : LineColor{255,92,128}
 C102 : Position{2,16}
 C103 : Curve{{31,3},{1,15},{32,4}}
 C104 : Position{32,16}
 C105 : Curve{{24,28},{26,30},{32,28}}
 C106 : TileEnd
 } 1606
 } 1604
 } 1607

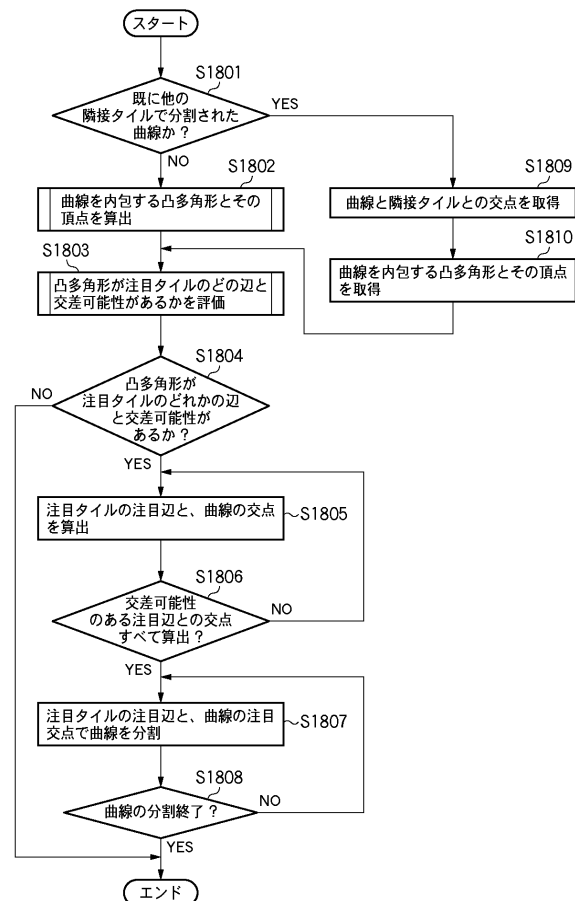
 C120 : TileStart (3,3)
 C121 : LineColor{255,92,128}
 C122 : Position{0,4}
 C123 : Curve{{6,2},{8,6},{0,16}}
 C124 : Position{0,28}
 C125 : Curve{{29,17},{1,29},{30,16}}
 C126 : TileEnd
 } 1608
 } 1605
 } 1609

 C500 : PageEnd

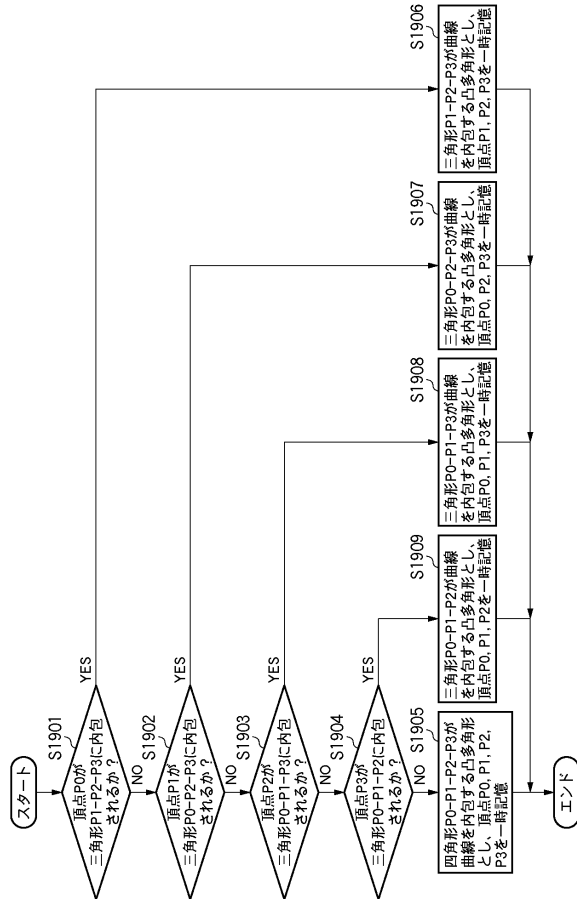
【図 16】



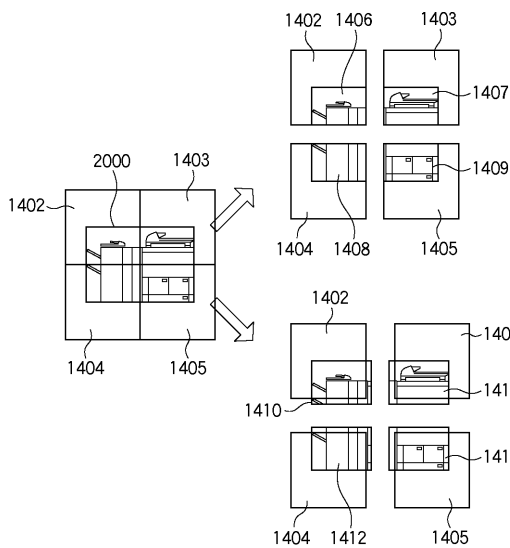
【図 17】



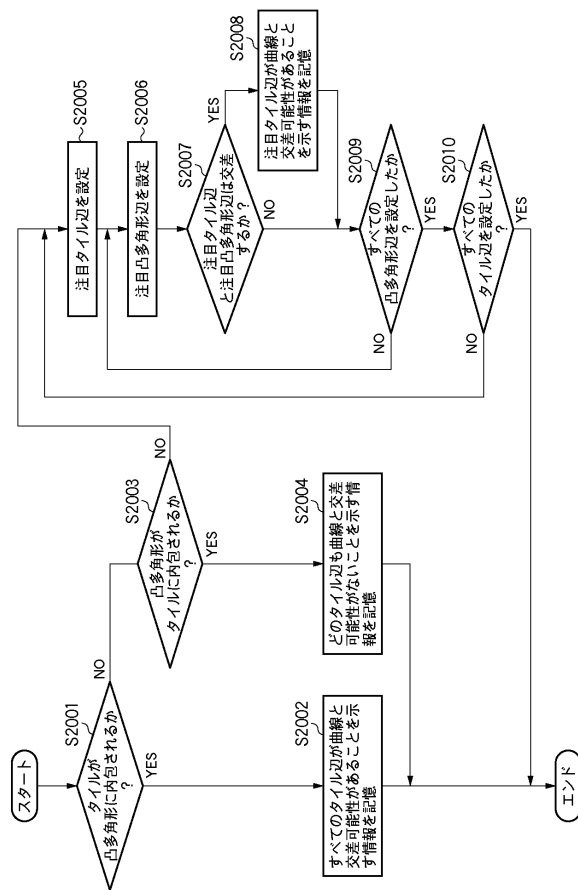
【図 18】



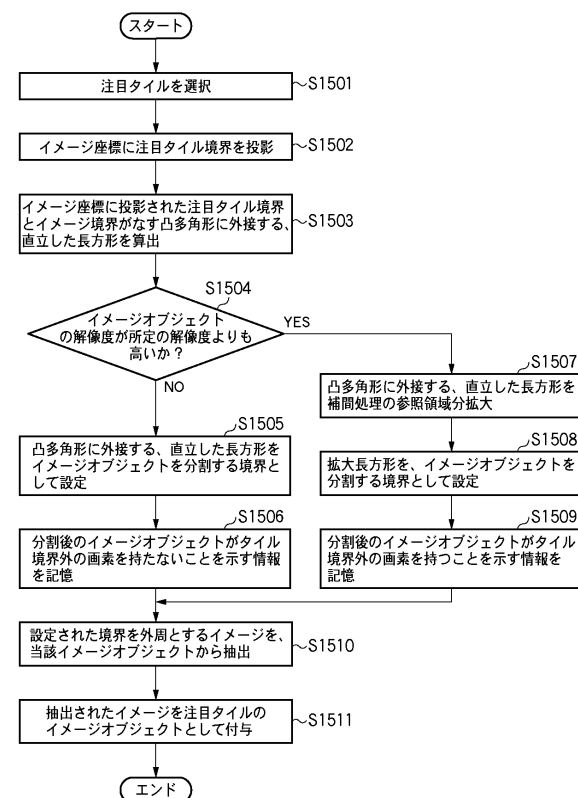
【図 20】



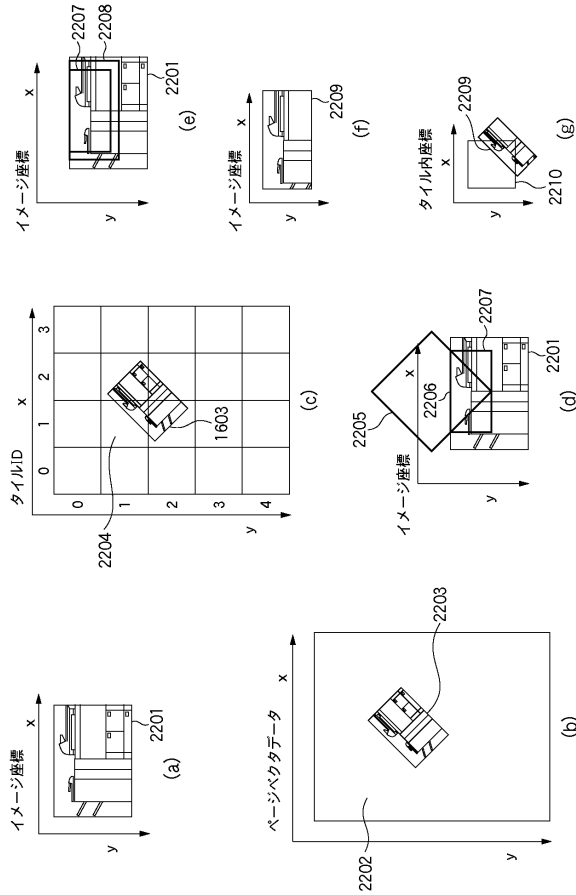
【図 19】



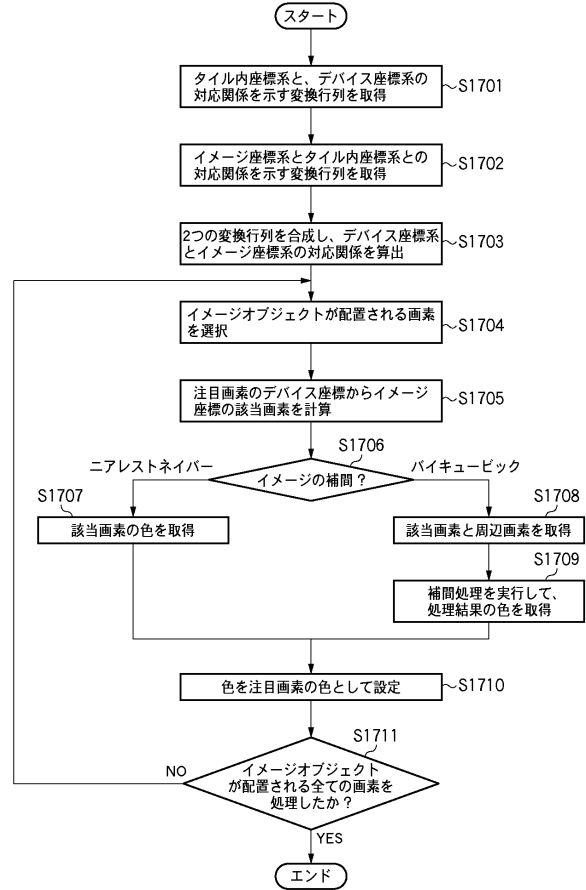
【図 21】



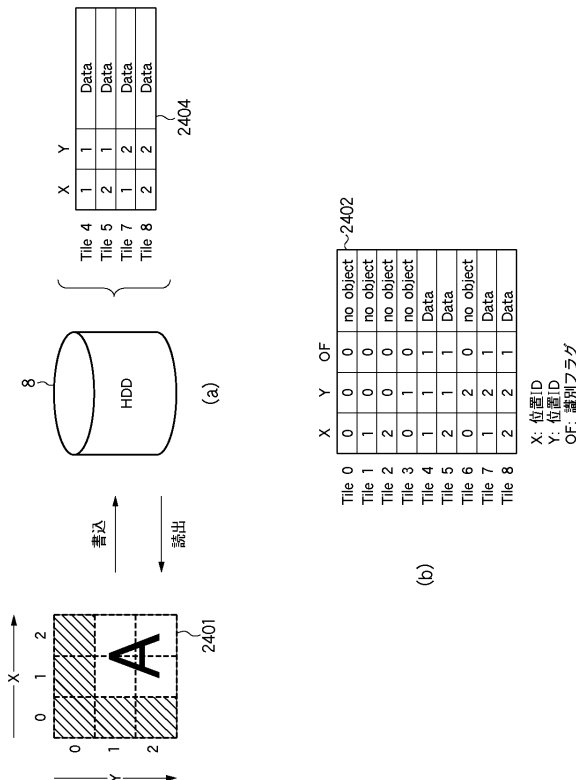
【図 2 2】



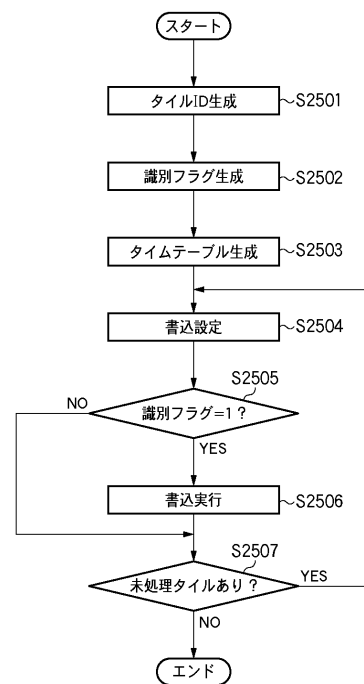
【図 2 3】



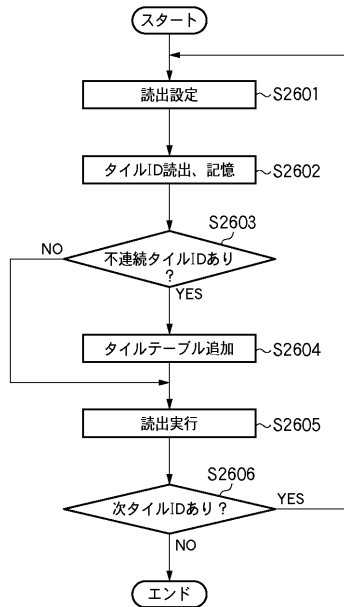
【図 2 4】



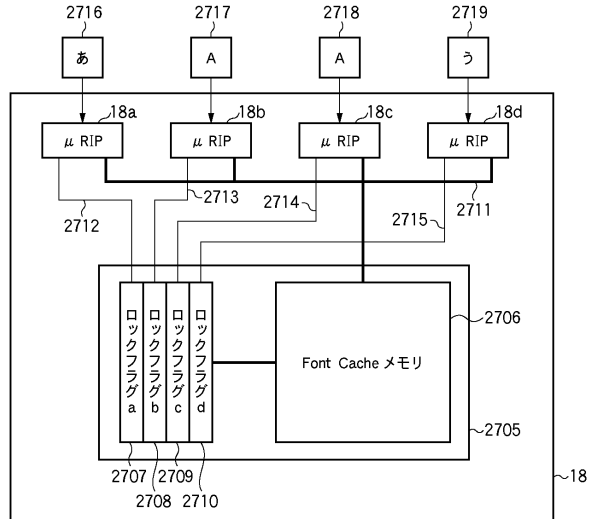
【図 2 5】



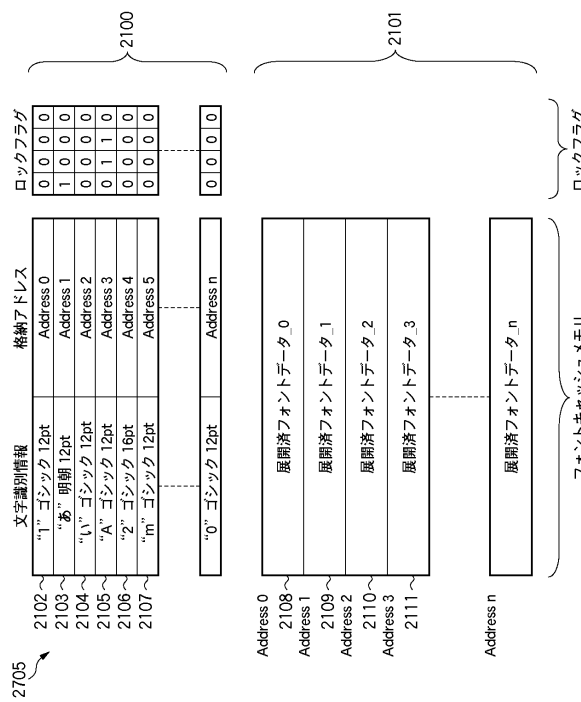
【図 26】



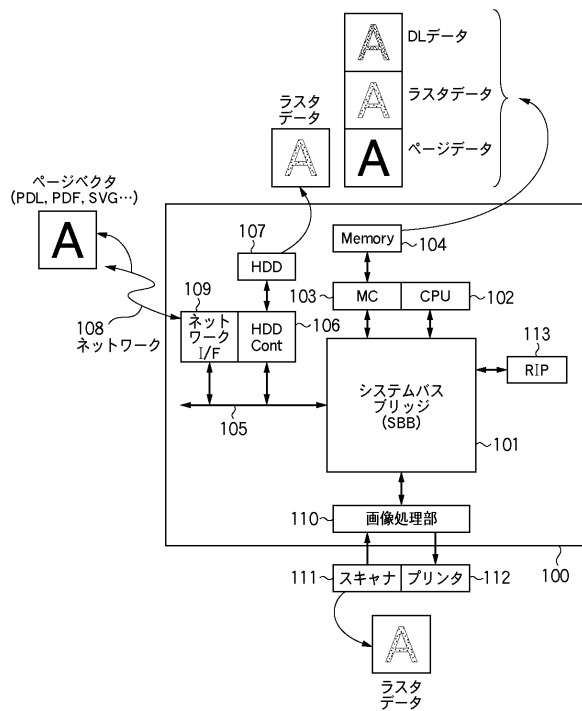
【図 27】



【図 28】



【図 29】



フロントページの続き

- (72)発明者 松井 信明
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 稲葉 恵司
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 前田 忠昭
東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

審査官 松永 稔

- (56)参考文献 特開平10-079942(JP,A)
特開平11-203061(JP,A)
特開昭59-047666(JP,A)
特開2004-272485(JP,A)
特開平07-073334(JP,A)
特開平10-307924(JP,A)
特開2006-309671(JP,A)
特開2006-54817(JP,A)

- (58)調査した分野(Int.Cl., DB名)
- | | |
|------|-------|
| H04N | 1/00 |
| B41J | 29/38 |