

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5188201号
(P5188201)

(45) 発行日 平成25年4月24日(2013.4.24)

(24) 登録日 平成25年2月1日(2013.2.1)

(51) Int.Cl.

F 1

G 0 6 F 3/12 (2006.01)

G 0 6 F 3/12

A

請求項の数 10 (全 23 頁)

(21) 出願番号	特願2008-43551 (P2008-43551)	(73) 特許権者	000001007
(22) 出願日	平成20年2月25日(2008.2.25)		キヤノン株式会社
(65) 公開番号	特開2009-199557 (P2009-199557A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成21年9月3日(2009.9.3)	(74) 代理人	100076428
審査請求日	平成23年2月22日(2011.2.22)		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、プログラム、記憶媒体

(57) 【特許請求の範囲】

【請求項 1】

プリンタドライバを有する画像処理装置であって、

前記プリンタドライバを用いて、アプリケーションが作成したアプリデータを解析して
画像のビットマップデータを生成する生成手段と、前記アプリデータに含まれる情報であって、当該アプリデータに基づいて前記生成手段
により生成されるビットマップデータには含まれない文字情報を取得する取得手段と、前記取得手段が取得した文字情報を、前記生成手段により生成されるビットマップデー
タの付加情報として、当該ビットマップデータに付加する付加手段と、前記付加手段により前記付加情報が付加されたビットマップデータ或いは前記生成手段
により生成されたビットマップデータを、プリンタに対して出力する出力手段と、前記ビットマップデータを保存する処理を前記プリンタに依頼する場合は、前記プリン
タにおいて、前記プリンタに保存されたビットマップデータの検索に利用するための前記
付加情報が付加されたビットマップデータを前記出力手段に出力させ、前記ビットマップ
データを印刷する処理を前記プリンタに依頼する場合は、前記付加情報が付加されてい
ないビットマップデータを前記出力手段に出力させる制御手段と、

を備えることを特徴とする画像処理装置。

【請求項 2】

前記付加情報は、前記アプリデータに含まれる文字コードの少なくとも一部を示す文字
情報であることを特徴とする請求項 1 に記載の画像処理装置。

10

20

【請求項 3】

前記付加情報は、前記アプリデータを生成したアプリケーションに関する文字情報であることを特徴とする請求項 1 に記載の画像処理装置。

【請求項 4】

前記付加情報は、前記アプリデータに含まれる少なくとも一部のオブジェクトの属性情報を示す文字情報であ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 5】

プリンタドライバを有する画像処理装置の制御方法であって、

前記プリンタドライバを用いて、アプリケーションが作成したアプリデータを解析して画像のビットマップデータを生成する生成工程と、

前記アプリデータに含まれる情報であって、当該アプリデータに基づいて前記生成工程で生成されるビットマップデータには含まれない文字情報を取得する取得工程と、

前記取得工程で取得した文字情報を、前記生成工程で生成されるビットマップデータの付加情報として、当該ビットマップデータに付加する付加工程と、

前記付加工程で前記付加情報が付加されたビットマップデータ或いは前記生成工程で生成されたビットマップデータを、プリンタに対して出力する出力工程と、

ビットマップデータを保存する処理を前記プリンタに依頼する場合は、前記プリンタにおいて、前記プリンタに保存されたビットマップデータの検索に利用するための前記付加情報が付加されたビットマップデータを前記出力工程で出力させ、ビットマップデータを印刷する処理を前記プリンタに依頼する場合は、前記付加情報が付加されていないビットマップデータを前記出力工程で出力させる制御工程と、
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 6】

前記付加情報は、前記アプリデータに含まれる文字コードの少なくとも一部を示す文字情報であることを特徴とする請求項 5 に記載の画像処理装置の制御方法。

【請求項 7】

前記付加情報は、前記アプリデータを生成したアプリケーションに関する文字情報であることを特徴とする請求項 5 に記載の画像処理装置の制御方法。

【請求項 8】

前記付加情報は、前記アプリデータに含まれる少なくとも一部のオブジェクトの属性情報を示す文字情報であることを特徴とする請求項 5 に記載の画像処理装置の制御方法。

【請求項 9】

請求項 5 から 8 のいずれか 1 項に記載の制御方法をコンピュータに実行させるためのプログラム。

【請求項 10】

請求項 5 から 8 のいずれか 1 項に記載の制御方法をコンピュータに実行させるためのプログラムを記憶したコンピュータ読み取り可能な記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、アプリケーションで作成したアプリデータに基づくビットマップデータを生成する画像処理装置及びその制御方法、プログラム、記憶媒体に関するものである。

【背景技術】

【0002】

ネットワークを介して接続された複数の装置（プリンタ、スキャナ、デジタル複写機、ファクシミリ装置）を連携して動作させることにより、高度な機能やより高い生産性を実現することが望まれている。また、これらの装置間で、解像度に依存しないデータ形式（以後、ベクタデータ）での画像のやり取りが考えられている。このベクタデータを受信した装置では、受信したベクタデータに基づいてビットマップイメージなどの画像データを生成（ラスターライズ）するため解像度の変換に伴う画像劣化が生じない。これにより各々

10

20

30

40

50

の装置において最適なビットマップイメージを生成して高品位な画像を得ることができるため、能力が異なる各種デバイスを相互に連携させるための技術として有効である。また更に、このベクタデータに加えて、実際に印刷される画像そのものとは異なる付加情報を保持し、画像を処理する際に参照したり、或いはその付加情報を基に画像を検索したりする技術も開発されている。

【 0 0 0 3 】

また、画像処理装置（ホストコンピュータ）から供給された画像データを出力装置の二次記憶装置にファイルとして保存しておけば、その出力装置を使用してユーザが好きな時間に、そのファイルを取り出して繰り返し印刷や出力を行うことができる。このように再利用を目的として出力装置の二次記憶装置に画像データをファイル形式で保存する機能をボックス機能、ファイルシステムをボックスと呼ぶ。このようなボックス機能を用いると、格納された画像データを再度印刷したり、能力の異なる他の装置に配信したりするなどの処理が容易になり、また一度生成した画像データを繰り返し再利用することも可能になる（特許文献 1 ～ 3 参照）。

【特許文献 1】特開平 1 1 - 3 3 1 4 5 5 号公報

【特許文献 2】特開平 8 - 6 3 3 4 5 号公報

【特許文献 3】特開 2 0 0 6 2 3 9 4 2 号公報

【発明の開示】

【発明が解決しようとする課題】

【 0 0 0 4 】

従来は、上述したような実際に印刷される画像そのものとは異なる付加情報は、アプリケーションが作成したアプリデータを展開することにより得られた画像データに対して、文字認識処理を施すことにより生成していた。そのため、アプリケーションにより作成されたアプリデータに含まれていた情報であって、当該アプリデータを展開することにより消失してしまった情報は、画像データを再利用する際に利用することができなかった。

【 0 0 0 5 】

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。

【 0 0 0 6 】

本願発明の特徴は、アプリケーションが作成したアプリデータに含まれる情報であって、当該アプリデータに基づいて生成されるビットマップデータには含まれない情報を利用

【課題を解決するための手段】

【 0 0 0 7 】

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、

プリンタドライバを有する画像処理装置であって、

前記プリンタドライバを用いて、アプリケーションが作成したアプリデータを解析して画像のビットマップデータを生成する生成手段と、

前記アプリデータに含まれる情報であって、当該アプリデータに基づいて前記生成手段により生成されるビットマップデータには含まれない文字情報を取得する取得手段と、

前記取得手段が取得した文字情報を、前記生成手段により生成されるビットマップデータの付加情報として、当該ビットマップデータに付加する付加手段と、

前記付加手段により前記付加情報が付加されたビットマップデータ或いは前記生成手段により生成されたビットマップデータを、プリンタに対して出力する出力手段と、

前記ビットマップデータを保存する処理を前記プリンタに依頼する場合は、前記プリンタにおいて、前記プリンタに保存されたビットマップデータの検索に利用するための前記付加情報が付加されたビットマップデータを前記出力手段に出力させ、前記ビットマップデータを印刷する処理を前記プリンタに依頼する場合は、前記付加情報が付加されていないビットマップデータを前記出力手段に出力させる制御手段と、を備えることを特徴とする。

【 0 0 0 8 】

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の制御方法は以下のような工程を備える。即ち、

プリンタドライバを有する画像処理装置の制御方法であって、

前記プリンタドライバを用いて、アプリケーションが作成したアプリデータを解析して画像のビットマップデータを生成する生成工程と、

前記アプリデータに含まれる情報であって、当該アプリデータに基づいて前記生成工程で生成されるビットマップデータには含まれない文字情報を取得する取得工程と、

前記取得工程で取得した文字情報を、前記生成工程で生成されるビットマップデータの付加情報として、当該ビットマップデータに付加する付加工程と、

前記付加工程で前記付加情報が付加されたビットマップデータ或いは前記生成工程で生成されたビットマップデータを、プリンタに対して出力する出力工程と、

ビットマップデータを保存する処理を前記プリンタに依頼する場合は、前記プリンタにおいて、前記プリンタに保存されたビットマップデータの検索に利用するための前記付加情報が付加されたビットマップデータを前記出力工程で出力させ、ビットマップデータを印刷する処理を前記プリンタに依頼する場合は、前記付加情報が付加されていないビットマップデータを前記出力工程で出力させる制御工程と、を備えることを特徴とする。

【 発明の効果 】

【 0 0 0 9 】

本発明によれば、アプリケーションが作成したアプリデータに含まれる情報であって、当該アプリデータに基づいて生成されるビットマップデータには含まれない付加情報を利用することができ、その付加情報をビットマップデータに付加するか否かにより、そのビットマップデータをプリンタにより印刷させるか、或いは保存させることができるようになる。更に、その付加情報を利用して、プリンタにおいて、そのプリンタで保存しているビットマップデータを検索することができるという効果がある。

【 発明を実施するための最良の形態 】

【 0 0 1 0 】

以下、添付図面を参照して本発明の好適な実施の形態を詳しく説明する。尚、以下の実施の形態は特許請求の範囲に係る本発明を限定するものでなく、また本実施の形態で説明されている特徴の組み合わせの全てが本発明の解決手段に必須のものとは限らない。

【 0 0 1 1 】

まず、本発明の第 1 の実施形態に係る画像処理装置（後述する）に接続された 1 D カラー系 MFP（多機能処理装置）の構成について説明する。

【 0 0 1 2 】

図 1 は、本実施の形態に係る MFP 1000 の構成を示す図である。

【 0 0 1 3 】

このカラー MFP 1000 は、原稿露光部、レーザ露光部、感光ドラム、作像部、定着部、給紙 / 搬送部及び、これらを制御するプリンタ制御部（不図示）を有している。

【 0 0 1 4 】

原稿露光部は、原稿台に置かれた原稿に照明を当てて原稿を光学的に読み取り、その像を電気信号に変換して画像データを作成する。レーザ露光部は、画像データに応じて変調されたレーザ光などの光線を等角速度で回転する回転多面鏡（ポリゴンミラー）に入射し反射走査光として感光ドラム 100 を照射する。作像部は、感光ドラム 100 を回転駆動するとともに、感光ドラム 100 の表面を帯電器によって帯電させ、レーザ露光部によって、その感光ドラム上に画像データに応じた静電潜像を形成する。そして、その静電潜像をトナーによって現像し、そのトナー像を転写ドラム 101 に巻回された記録媒体（シート）に転写する。その際に転写されずに感光ドラム 100 上に残った微小トナーを回収するといった一連の電子写真プロセスを実行してシートに像を形成する。ここでシートが転写ドラム 101 に巻き付けられて 4 回転する間に、マゼンタ（M）、シアン（C）、イエロー（Y）、ブラック（K）のトナーを持つ現像ユニット 102 ~ 105 が入れ替わって

トナーを供給して順次前述の電子写真プロセスを繰り返し実行する。こうして転写ドラム 101 が 4 回転して 4 色のフルカラートナー像が転写されたシートは転写ドラム 101 を離れて定着部へ搬送される。

【0015】

定着部は、ローラやベルトの組み合わせによって構成され、ハロゲンヒータなどの熱源を内蔵し、作像部によってトナー像が転写されたシート上のトナーを、熱と圧力によって溶解、定着させる。

【0016】

給紙 / 搬送部は、シートカセットやペーパーデッキに代表されるシート収納庫を一つ以上有しており、プリンタ制御部の指示に応じてシート収納庫に収納された複数のシートの中から一枚を分離して作像部・定着部へ搬送する。シートは作像部の転写ドラム 101 に巻きつけられ、4 回転した後に定着部へ搬送される。このとき上述したように 4 回転する間に前述の Y M C K 各色のトナー像がシートに転写される。またシートの両面に画像形成する場合は、定着部を通過したシートを再度作像部へ搬送する搬送経路を通るように制御する。

10

【0017】

プリンタ制御部は、MFP 全体を制御する MFP 制御部と通信して、その指示に応じて制御を実行すると共に、前述のスキナ、レーザ露光、作像、定着、給紙 / 搬送の各部の状態を管理しながら、全体が調和を保って円滑に動作できるよう指示を行う。

【0018】

20

< MFP 制御部の構成 >

図 2 は、MFP 1000 の MFP 制御部の構成例を説明するブロック図である。

【0019】

MFP 制御部 200 は、画像入力デバイスであるスキナ 201 や画像出力デバイスであるプリンタエンジン 202 と接続し、原稿上の画像の読み取りや印刷などの制御を行う。尚、スキナ 201 は前述の原稿露光部に相当し、プリンタエンジン 202 は、図 1 のレーザ露光部、感光ドラム、作像部、定着部、給紙 / 搬送部を含む。この MFP 制御部 200 は、LAN 10 や公衆回線 204 と接続することで画像データやデバイス情報を LAN 10 経由で入出力するための制御を行う。

【0020】

30

CPU 205 は、この MFP 全体を制御するための中央処理装置である。RAM 206 は、CPU 205 の処理を実行時に各種データを一時的に記憶するワークメモリを提供するとともに、入力された画像データを一時記憶するメモリでもある。ROM 207 はブート ROM であり、ブートプログラムが格納されている。HDD 208 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や、入力された画像データ等を格納する。操作部インタフェース (I/F) 209 は、画像データ等を表示可能な表示部を有する操作部 210 とのインタフェースを制御するインタフェース部であり、操作部 210 に対して表示データを出力する。また操作部 I/F 209 は、操作部 210 のキーなどを使用して操作者が入力した情報を CPU 205 に伝える役割をする。ネットワーク I/F 211 は、例えば LAN カード等で実現され、LAN 10 を介して情報の入出力を行う。またモデム 212 は公衆回線 204 に接続され、公衆回線 204 に接続された外部装置との間で情報の入出力を行う。以上の各部がシステムバス 213 上に配置されている。

40

【0021】

イメージバス I/F 214 は、システムバス 213 と、画像データを高速で転送する画像バス 215 とを接続するためのバス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 215 上には、ラスタイメージプロセッサ (RIP) 216、デバイス I/F 217、スキナ画像処理部 218、プリンタ画像処理部 219、画像処理部 220、カラーマネージメントモジュール (CMM) 230 が接続される。ラスタイメージプロセッサ (RIP) 216 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画像データに展開する。デバイス I/F 部 217 は、スキナ 201 やプリ

50

ンタエンジン 202 と MFP 制御部 200 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 218 は、スキャナ 201 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。プリンタ画像処理部 219 は、印刷する画像データに対して、プリンタエンジン 202 に応じた補正、解像度変換等の処理を行う。画像処理部 220 は、画像データの回転や、画像データの圧縮伸長処理等の各種画像処理を行う。CMM 230 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（色空間変換処理ともいう）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 L a b など）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とは、カラー MFP におけるスキャナ 201 やプリンタエンジン 202 の色再現特性を修正するためのデータである。

10

【0022】

<システム構成>

図 3 は、MFP 1000 と同様の機能を備えた複数の MFP からなる画像処理システムの全体構成を示すブロック図である。

【0023】

図において、この画像処理システムは、互いに LAN 10 を介して接続された、MFP 1, MFP 2, MFP 3 を具備している。ここで各 MFP は、前述の図 2 に示すような構成を有している。HDD（二次記憶装置）H1, H2, H3 は図 2 の HDD 208 に相当している。ここで各 MFP に搭載されているプリンタエンジン 202 の印刷解像度は、MFP 毎に異なっており、MFP 1 と MFP 3 はともに 600 dpi、MFP 2 は 1200 dpi とする。また各 MFP に搭載されている RIP 216 の種類も MFP 毎に異なっており、MFP 1 と MFP 2 の RIP は同種（図中、「Ra」）、MFP 3 だけが異なる種類（「Rb」）である。一般に RIP 216 は、ASIC などのハードウェアで構成されるため、異なる種類のレンダラは異なる種類の描画命令群を処理することができない。この描画命令群を一般に DisplayList（以下、DL）と呼ぶ。DL はハードウェアで処理可能なインストラクションであり、複雑な描画記述を持つベクタデータからソフトウェアで生成され、解像度に依存している。

20

【0024】

ここで MFP 1, MFP 2, MFP 3 は、ネットワークプロトコルを使用して互いに通信することができる。尚、LAN 10 に接続されるこれらの MFP は上記のような物理的な配置に限定されなくても良い。また、LAN 10 には MFP 以外の機器（例えば PC、各種サーバ、プリンタなど）が接続されていても良い。

30

【0025】

図 4 は、MFP 1000 の動作を制御する MFP 制御部のソフトウェアの構成を示す機能ブロック図である。

【0026】

プリンタインターフェイス 400 は、LAN 10 を介して外部装置（ホストコンピュータ等）とのデータの送受信を制御する。プロトコル制御部 401 は、ネットワークプロトコルを解析・送信することによって外部装置との通信を行う。ベクタデータ生成部 402 は、ビットマップデータから解像度に依存しない描画記述であるベクタデータを生成（ベクタライズ）する。メタデータ生成部 403 は、ベクタライズの過程で得られる副次情報をメタデータとして生成する。ここでメタデータとは、画像を描画する処理には必要のない検索用の付加的なデータを指す。PDL 解析部 404 は、PDL を解析し、より処理しやすい形式の中間コード（DisplayList）に変換する。PDL 解析部 404 で生成された中間コードはデータ描画部 405 に渡されて処理される。データ描画部 405 は、中間コードをビットマップデータに展開する。こうして展開されたビットマップデータは、RAM 206 に設けられたページメモリ 406 に逐次描画される。パネル入出力制御部 407 は、操作部 210 との間のデータのやり取りを制御する。ドキュメント記憶部 408 は、入力文書の一塊（ジョブ）単位にベクタデータ、DisplayList、メタデータ等を包含する

40

50

データファイルを記憶する。このドキュメント記憶部 408 は、HDD 208 等の二次記憶装置によって実現される。尚、このデータファイルを本実施の形態では「ドキュメント（またはDocument）」と呼ぶ。スキャン制御部 409 は、スキャナ 201 から入力した画像データに対して、補正、加工、編集などの各種処理を行う。印刷制御部 410 は、ページメモリ 406 の内容をビデオ信号に変換処理してプリンタエンジン 202 へ転送する。

【0027】

< MFP におけるデータ処理の説明 >

次に、ドキュメントを構成するベクタデータ、DL、メタデータがどのように生成されるのかを説明する。

【0028】

図5～図7は、MFP 1000 のMFP 制御部によるデータ処理の流れを説明する図である。

【0029】

図5は、MFP 1000 がコピー動作を実行するときのデータフローを説明する図である。

【0030】

まず原稿露光部にセットされた原稿はスキャン処理 501 によってビットマップデータに変換される。次にベクタライズ処理 502 とメタデータ生成処理 504 によってビットマップデータからそれぞれ解像度に依存しないベクタデータと、それに付随するメタデータが生成される。これらベクタデータ、メタデータの具体的な生成方法については後述する。

【0031】

次にドキュメント生成処理 503 によってベクタデータとメタデータが関連付けられたドキュメントが生成される。次にDL生成処理 505 によりドキュメント中のベクタデータからDLが生成され、この生成されたDLはドキュメント中に格納されると共にレンダリング処理 507 に送られてビットマップデータに展開される。こうして展開されたビットマップデータは、印刷処理 508 によって記録媒体であるシートに印刷されて印刷物となる。尚、こうして印刷された印刷物をまた原稿露光部にセットしてスキャン処理 501 からの処理を行うことができる。

【0032】

図6は、図5で示したメタデータ生成処理の具体的な処理の流れを示す図である。

【0033】

まず領域分割処理 601 により、ビットマップデータから領域分割を行う。この領域分割は、入力されたビットマップ画像データを解析して、画像に含まれるオブジェクトの塊毎に領域に分割し、各領域の属性を判定して分類する処理である。ここで属性としては、文字 (TEXT)、画像 (PHOTO)、線 (LINE)、図形 (PICTURE)、表 (TABLE) 等がある。

【0034】

図8は、入力画像に対して領域分割を行った場合の一例を示す図である。

【0035】

入力画像 801 に対して領域分割を行った結果が判定結果 802 で示されている。この判定結果 802 において、点線で囲った部分が画像を解析した結果のオブジェクトの1単位を表し、各オブジェクトに対して付されている属性の種類が領域分割の判定結果を示している。

【0036】

こうして属性毎に分類された各領域の中から文字属性の領域は、文字認識処理 602 により文字認識処理され、その領域の文字列は文字コードに変換される。

【0037】

一方、画像属性の領域は、画像情報抽出処理 603 により画像情報に変換される。この画像情報とは、その画像の特徴を現す文字列であり、例えば「花」や「顔」といった文字列で表される。この画像情報の抽出には画像特徴量 (画像を構成するピクセルの周波数や

10

20

30

40

50

濃度など)の検出や顔認識などの一般的な画像処理技術を用いて実現できる。こうして生成された文字列と画像情報は、フォーマット変換処理604によって後述するデータフォーマットに整えられてメタデータが生成される。

【0038】

図7は、PDLプリント時のデータ処理の流れを示す図である。ここでPDLプリントとは、PC(ホストコンピュータ)上のアプリケーションから印刷を指示した場合にPC上のプリンタドライバによって生成されたページ記述言語(PDL)を受け取って印刷するプリンタの処理動作である。

【0039】

PCから受信したPDLデータは、PDLデータ解析処理701によって解析され、ベクタデータが生成される。次にDL生成処理702によりベクタデータからDLが生成され、生成されたDLはドキュメントの中に格納されると共にレンダリング処理703に送られてビットマップデータに展開される。こうして展開されたビットマップデータは印刷処理704によってシートに印刷されて印刷物となる。この過程で生成されるベクタデータ、DLは、ドキュメント生成処理706によってドキュメントとして格納される。

【0040】

またレンダリング処理703が生成したビットマップデータから、メタデータ生成処理705により、図5のコピー動作時と同様に、文字列や画像情報がメタデータとして生成されてドキュメントに格納される。更に、PDLの依存性を除去するために、ベクタデータ生成処理707の様に、レンダリング後のビットマップデータからベクタデータを生成する場合もある。

【0041】

なお、ここではMFP1000は文字フォント情報を保持していないものとする。そのため、PC上のアプリケーションで作成されたアプリデータに文字情報が含まれている場合は、後述するプリンタドライバで予め文字情報が画像データに展開された状態でPDLデータとしてMFP1000に送信される。

【0042】

次に、ドキュメントの生成処理と印刷処理について説明する。

【0043】

図9は、MFP制御部によるドキュメントの生成処理を示すフローチャートである。この処理はビットマップデータを受けてベクタデータ、DL、メタデータで構成されるドキュメントを生成する処理である。この処理は、HDD208からRAM206にロードされたプログラムに従ってCPU205の制御の下に実行される。

【0044】

まずステップS1で前述した領域分割処理を行う。次にステップS2に進み、その分割した各領域の種別(属性)を判定する。ここでは前述のように、文字、画像、図形とからなる属性ごとに分類し、各属性に対して別々の処理を行う。尚、図8では、属性を文字、画像、線、図形、表に分類した例を示したが、図9では、画像及び図形は「IMAGE」に、線及び表は「GRAPHIC」に分類される。尚、文字は図8と同様に同じ「TEXT」に分類される。

【0045】

属性が「TEXT」の場合はステップS2からステップS3に進み、その文字画像に対して文字認識処理を行った後、ステップS4に進んで文字列を取得する。その後、ステップS5に進み、ステップS4で取得した文字列をメタデータに変換する。次にステップS6に進み、その認識した文字の輪郭をベクタデータに変換する。そしてステップS11で、ドキュメントのフォーマットに変換してドキュメントに含める。

【0046】

ここで、ステップS5の処理に関してもう少し説明を加える。

【0047】

文字列から生成されるメタデータは、文字コードの羅列であるが、文字コードの羅列は

10

20

30

40

50

キーワードによる検索に必要な情報である。しかし、文字認識処理では、文字画像から文字コードへの変換ができて、例えば、その文字画像の「明朝」「ゴシック」等の書体や、「10pt」「12pt」といった文字のサイズ、「イタリック」「ボールド」といった文字修飾までは認識できない。従って、その文字認識された文字コードから文字画像を描画する場合には、文字コードではなく、その文字輪郭をベクタデータとして保持する必要がある。

【0048】

一方、ステップS2で、その領域の属性がIMAGEであると判定した場合はステップS7に進み、画像情報の抽出処理を行う。ステップS7では、前述したように、画像特徴量検出や顔認識などの一般的な画像処理技術を用いて画像の特徴を検知する。次にステップS8に進み、ステップS7で検知した画像の特長を文字列に変換する。この変換は特徴パラメータと、それに対応する文字列を記憶したテーブルを用いることにより容易に実施できる。その後、ステップS9に進み、その文字列をメタデータへ変換する。そしてステップS11に進み、ドキュメントのフォーマットに変換してドキュメントに含める。このように属性がIMAGEである領域に対してはベクトル化を行わずにイメージデータをそのままベクタデータとして保持する。

10

【0049】

またステップS2で、その領域の属性がGRAPHICであると判定した場合はステップS10に進み、ベクトル化処理を行う。そしてステップS11に進んで、ドキュメントのフォーマットに変換してドキュメントに含める。

20

【0050】

図10は、ドキュメントの印刷処理を示している。この処理は生成されたドキュメントを印刷出力する処理である。

【0051】

図10は、本実施の形態に係るMFP制御部によるドキュメントの印刷処理を示すフローチャートである。この処理は、HDD208からRAM206にロードされたプログラムに従ってCPU205の制御の下に実行される。

【0052】

まずステップS21で、ドキュメントを受け取る。次にステップS22に進み、そのドキュメントのベクタデータからDLを生成する。次にステップS23に進み、ステップS22で生成したDLをドキュメントに追加する。次にステップS24に進み、DLをビットマップデータにレンダリングする。最後にステップS25に進み、そのビットマップデータに基づいてシートへの印刷処理を行う。

30

【0053】

図11は、本実施の形態に係るMFP制御部によるPDLデータからドキュメントを生成して印刷する処理を示すフローチャートである。この処理は、HDD208からRAM206にロードされたプログラムに従ってCPU205の制御の下に実行される。

【0054】

まずステップS31で、PDLデータを解析する。ステップS32では、この解析中に文字列情報などのメタデータ（付加情報）がPDLデータに含まれているかを判定する。メタデータが含まれていればステップS39に進み、そのPDLデータのメタデータをドキュメントに追加する。

40

【0055】

一方、ステップS32で、文字列情報などのメタデータ以外のデータの場合はステップS33に進み、ベクタデータに変換する。そしてステップS34に進み、ドキュメントを生成する。次にステップS35に進んでDLを生成する。次にステップS36に進み、ステップS35で生成したDLをドキュメントに追加する。ここまでの処理フローでドキュメントが生成される。そしてステップS37のレンダリング処理、ステップS38のシートへの印刷処理により全ての処理を終了する。

【0056】

50

<ドキュメントデータ構造>

次に、ドキュメントのデータ構造を説明する。

【0057】

図12、図13、図14はドキュメントの構造を示している。

【0058】

図12は、本実施の形態に係るドキュメントのデータ構造を示す図である。

【0059】

ドキュメントは、複数ページからなるデータであり、大きく分けるとベクタデータ1200、メタデータ1201、DL1202で構成され、ドキュメントヘッダ1203を先頭とする階層構造である。ベクタデータ1200は更に、ページヘッダ1204、サマリ情報1205、オブジェクト1206を含んでいる。またメタデータ1201は、ページ情報1207と詳細情報1208とを含んでいる。DL1202は、ページヘッダ1209と描画展開用のインストラクション1210を有している。ドキュメントヘッダ1203には、ベクタデータの格納場所とDLの格納場所が記述されている。これにより、ベクタデータ1200とDL1202はドキュメントヘッダ1203によって関連付けられている。

10

【0060】

ベクタデータ1200は、解像度非依存の描画データである。ページヘッダ1204は、各ページの大きさや向きなどのレイアウト情報を記述する。そしてオブジェクト1206には、ライン、多角形、ベジェ曲線などの描画データが一つずつリンクされており、複数のオブジェクトがまとめてサマリ情報1205により関連付けられている。このサマリ情報1205は、複数のオブジェクトの特徴をまとめて表現するものであり、図8で説明した各領域の属性情報などが記述される。

20

【0061】

メタデータ1201は、画像を描画する処理には直接関係しない検索用の付加情報である。ページ情報1207は、例えばメタデータがビットマップデータから生成されたものであるか、PDLデータから生成されたものであるか、等のページ情報を記述する。また詳細情報1208は、OCR情報や、画像情報として生成された文字列（文字コード列）を記述する。

【0062】

またベクタデータ1200のサマリ情報1205はメタデータを参照しており、このサマリ情報1205から、対応する詳細情報1208を見つけることができる。

30

【0063】

DL1202は、レンダラ(RIP216)がビットマップデータに展開するための中間コードである。ページヘッダ1209には、ページ内の描画情報(インストラクション)の管理テーブルなどが記述されている。またインストラクション1210は、解像度依存の描画情報で構成されている。

【0064】

図13は、ドキュメントデータの具体例を示す図である。

【0065】

図では、1ページ目のサマリ情報1301、1302には、「TEXT」と「IMAGE」が保持されている。「TEXT」のサマリ情報には、「H,e,l,l,o」(オブジェクト1303)と「W,o,r,l,d」(オブジェクト1304)の文字輪郭がベクタデータとしてリンクされている。更に、サマリ情報1301から、「Hello」「World」という文字コード列(メタデータ)1305が参照されている。また「IMAGE」のサマリ情報1302には、蝶の写真画像(JPEG)1306がリンクされている。そしてサマリ情報1302から「butterfly」という画像情報(メタデータ)1307が参照されている。従って、例えば「World」というキーワードでページ中のテキストを検索する場合は、以下の手順で行う。

40

【0066】

まずドキュメントヘッダ1203からベクタデータのページデータを順次取得し、ペー

50

ジヘッダ 1 2 0 4 にリンクされているサマリ情報 1 3 0 1 から属性「TEXT」にリンクされているメタデータ 1 3 0 5 を検索する。

【 0 0 6 7 】

図 1 4 (A) (B) は、例えば図 1 2 で説明したデータ構造が、メモリ或はファイルにどのように配置されるのかを示す図である。

【 0 0 6 8 】

図 1 4 (A) は、メモリへのドキュメントの配置を説明する図である。ここではドキュメントは、ベクタデータ領域、メタデータ領域、D L 領域がメモリ上の任意のアドレスに配置される。

【 0 0 6 9 】

図 1 4 (B) は、ファイルへのドキュメントの配置を説明する図である。ここではドキュメントは、ベクタデータ領域、メタデータ領域、D L 領域が、一つのファイルにシリアルライズされる。

【 0 0 7 0 】

以上、M F P 1 0 0 0 が有しているソフトウェアにより P D L データを解析して、ビットマップデータに展開し、印刷する処理について説明した。ところで、上述したように、M F P が文字フォント情報を保持していない場合は、文字情報についてはホストコンピュータ側のプリンタドライバで予め画像データに展開された状態で P D L データとして M F P に送信する必要がある。しかしながらこの場合、M F P が受信した P D L データに基づいてベクタデータを生成するとともに、上述した方法でメタデータを生成しようとする

と、ビットマップデータに対して文字認識処理を行って文字列情報を取得しなければならない。なぜなら、ホストコンピュータ側のアプリデータに含まれていた文字情報は、プリンタドライバによって画像データに展開された状態で M F P に送信されるため、文字列情報は消失してしまっているからである。従って、この M F P における文字認識処理で文字が誤認識されてしまうと、ドキュメントのメタデータとして保持される文字列情報が示す文字と実際の画像に含まれている文字とが異なってしまうおそれがある。

【 0 0 7 1 】

そこで、本実施の形態では、M F P に接続された画像処理装置（例えば、パーソナルコンピュータ（P C））において、アプリケーションが作成したアプリデータを展開して画像データを生成する際に以下のような処理を行う。即ち、本実施の形態の画像処理装置（P C）は、アプリケーションが作成したアプリデータを展開して画像データを生成する場合に、アプリデータに含まれる情報であって、当該アプリデータに基づいて生成された画像データには含まれない情報を取得する。そして、取得した情報を生成された画像データの付加情報として、当該画像データに付加する。更に、この付加情報が付加された画像データを外部装置（M F P）に送信する。なお、下記に説明するアプリケーションとは、文書作成や表計算などの用途に用いられるアプリケーションを示す。

【 0 0 7 2 】

また、ここでは、ホストコンピュータ（P C）と印刷装置（プリンタ）とがネットワークを介して接続されている場合について説明する。なお、ホストコンピュータと印刷装置との接続形態はここで説明するネットワークの形態に限らず、セントロニクスインタフェース、赤外線通信、無線通信、U S B や I E E E 1 3 9 4 規格で定義されたインタフェース等を採用しても構わない。

【 0 0 7 3 】

図 1 5 は、本実施の形態において画像処理装置として機能する P C 1 5 5 0 からプリンタ 1 5 2 0 に印刷データを送信して印刷させる印刷システムにおけるプリンタドライバ 1 5 0 1 の機能構成を説明する図である。図において、C P U 1 5 3 1 は R O M 1 5 3 5 に格納されているブートプログラムに基づいて P C 1 5 5 0 のシステムを起動する。そして C P U 1 5 3 1 は、このシステム上で R O M 1 5 3 5 に格納されている各種制御プログラムを読み出して、R A M 1 5 3 2 をワークエリアとして各種処理を実行する。操作部 1 5 3 3 は、ユーザからの指示を受け付け、その指示内容を C P U 1 5 3 1 に通知する。ネッ

10

20

30

40

50

トワーク I / F 部 1 5 3 4 は、P C 1 5 5 0 とプリンタ 1 5 2 0 等のネットワーク上の外部装置（例えば M F P ）との間のデータ通信を制御する。

【 0 0 7 4 】

更に、図において、1 5 0 0 は、P C 1 5 5 0 を制御している O S を示す。プリンタドライバ 1 5 0 1 は、O S 上で動作する、プリンタ 1 5 2 0 を制御するソフトウェアである。1 5 0 2 は、P C 1 5 5 0 に備えられたアプリケーションを用いて作成されたアプリデータに基づく画像データをプリンタ 1 5 2 0 に送信して印刷させる指示や、或いは保存させる指示を受け付けるためのインタフェースを、操作部 1 5 3 3 を介して提供する。印刷データ制御部 1 5 0 4 は、アプリケーションから指定される描画命令を受け、プリンタ 1 5 2 0 で印刷できる印刷データを作成する。アプリ情報取得部 1 5 1 0 は、アプリデータを
10 作成するアプリケーションに関するアプリ情報を取得する。このアプリ情報とはアプリケーションの製品名やバージョンなどの情報である。ホスト情報取得部 1 5 1 2 は、設定情報を取得する。この設定情報とは、ユーザインタフェース部 1 5 0 2 で設定される情報であって、プリンタ 1 5 2 0 で印刷処理を実行する際の用紙の向き、サイズ、部数、両面 / 片面指定などを含む。描画命令取得部 1 5 1 1 は、描画命令を取得する。この描画命令は、線描画やビットマップ描画などの命令で様々なパラメータを含む。また描画に必要なデータが O S のサービス関数によって得られる場合もあり、このサービス関数によって得られる情報はプリンタ 1 5 2 0 におけるベクタライズの処理においても利用される。印刷
20 データ生成部 1 5 1 4 は、アプリデータを展開することにより画像データを生成する。前述したように、M F P 側に文字フォント情報が保持されていない場合、アプリデータに含まれる文字情報はプリンタドライバによって画像データに展開される。なお、ここではアプリデータに含まれる文字情報以外の情報は画像データに展開せずに P D L に従って記述されて M F P に送信するものとするが、文字情報以外の情報についても画像データに展開した状態で送信するようにしても構わない。メタデータ生成部 1 5 1 3 は、アプリ情報取得部 1 5 1 0、ホスト情報取得部 1 5 1 2、描画命令取得部 1 5 1 1 で取得した情報を基にメタデータを生成する。即ち、メタデータ生成部 1 5 1 3 は、アプリデータに含まれる情報であって、当該アプリデータに基づいて生成された画像データには含まれない情報を取得し、この情報に基づいてメタデータを生成する。印刷コマンド制御部 1 5 0 3 は、印刷データ生成部 1 5 1 4 により作成された印刷データ及びメタデータ生成部 1 5 1 3 により生成されたメタデータをプリンタ 1 5 2 0 に対応した印刷コマンドに変更する。
30

【 0 0 7 5 】

メタデータ生成部 1 5 1 3 により生成されたメタデータは、印刷データ生成部 1 5 1 4 により生成された画像データの付加情報として、当該画像データに付加される。そして、付加情報が付加された画像データは、ネットワーク I / F 部 1 5 3 4 を介してプリンタ 1 5 2 0 に対して出力される。

【 0 0 7 6 】

なお、プリンタドライバ 1 5 0 1 のメタデータ生成部 1 5 1 3 で作成されたメタデータと、上述の M F P 1 0 0 0 のメタデータ生成部 4 0 3 で作成されたメタデータは、M F P 上でマージされる。即ち、プリンタドライバ 1 5 0 1 のメタデータ生成部 1 5 1 3 で作成されたメタデータは、M F P 1 0 0 0 のメタデータ生成部 4 0 3 で作成されたメタデータ
40 とともに、「検索」や「ベクタライズ」に利用される。

【 0 0 7 7 】

図 1 6 は、M F P 1 0 0 0 において、プリンタドライバが生成したメタデータを用いて、P D L データに基づくドキュメントを生成して印刷する処理の流れを説明する図である。ここでは前述の図 7 と共通する部分は同じ記号で示し、それらの説明を省略している。

【 0 0 7 8 】

ここでは、上述のプリンタドライバ 1 5 0 1 で生成されたメタデータ 1 6 0 0 が、P D L データをレンダリングすることにより得られたビットマップデータから生成されたメタデータとマージされドキュメントが生成される。このメタデータは、このドキュメントの検索等に利用される。また、M F P において、プリンタドライバ 1 5 0 1 で生成されたメ
50

タデータ 1600 を参照してベクタライズすることにより、元のアプリデータにより近い内容を示すベクタデータを得ることができる。

【0079】

図17は、プリンタドライバによるメタデータの生成処理を説明する図である。

【0080】

アプリケーション1700からのアプリデータやホスト情報1701からプリンタドライバ1501により、PDLデータ1702とメタデータ1703が生成される。即ち、プリンタドライバ1501がアプリデータを展開して画像データを生成する時、「ジョブの処理方法」が「保存」の場合、アプリデータに含まれている情報を取得し、メタデータ1703を生成する。そして、プリンタ1520に、PDLデータ1702と共にメタデータ1703を送信する。なお、このときメタデータ1705を、描画データ1704とマージして、PDLデータの中に格納するようにしてもよい。このようにして、プリンタ1520のボックスに保存されたドキュメントを検索する際に、プリンタドライバ1501で生成されたメタデータを利用することが可能になる。

10

【0081】

以上の通り、プリンタによっては文字フォント情報を保持していないものもあり、このような場合はホストコンピュータ側で予めアプリデータを画像データに展開した状態でプリンタに送信する。しかしながら、この場合アプリデータに基づいて画像データを生成する際に、文字列情報が失われてしまう場合がある。このため、プリンタ側でビットマップデータに対して文字認識処理を行った結果、ビットマップデータに含まれている文字列を正しく認識できなかった場合、それを基に作成したメタデータを用いて検索すると検索に失敗する場合があった。これに対して、本実施の形態では上述のように構成することで、例えば「ジョブの処理方法」が「保存」の場合（後に検索される可能性が高い）には以下のようにする。即ち、アプリデータを展開して画像データを生成する時に、そのアプリデータに含まれる文字情報（文字コードの少なくとも一部、展開後の画像データには含まれない）を取得し、メタデータ（付加情報）として付加する。そして、そのメタデータを、プリンタにおけるドキュメントの検索等に利用する。

20

【0082】

尚、本実施の形態では、「ジョブの処理方法」が「印刷」の場合は、メタデータを付与せずに、印刷に要する時間を短縮するようにしている。

30

【0083】

〔実施の形態2〕

上述の実施の形態1では、プリンタドライバで作成したメタデータをMFP1000に送信して、そのメタデータをMFP1000における検索で用いる例を説明した。本実施の形態2ではMFP1000でベクタライズの処理を行う際にプリンタドライバで生成したメタデータを利用する例を説明する。

【0084】

プリンタの中には、画像の透過やグラデーションの属性に対応していないものもあるため、これらのプリンタに対しては予め該当するオブジェクトを画像データに展開した状態でプリンタに送信する必要がある。このため、アプリデータを展開して画像データを生成する際に、それらの画像の透過やグラデーションの属性に関する情報が消失してしまう。これにより、透過やグラデーション等の属性を有していたオブジェクトを含むアプリデータに基づいて生成された画像データを、プリンタでベクタライズする際に、透過イメージの重なり部分が分離されて隙間が生じる場合がある。

40

【0085】

図18は、本発明の実施の形態2を説明する図である。

【0086】

図において、アプリケーション1800で作成されたアプリデータ1801には、円形のオブジェクトと四角形のオブジェクトとが重なり合った領域が50%透過であることを示すグラフィックスの属性情報が含まれている。このアプリデータ1801をプリンタド

50

ライバ１５０１が画像データに展開すると、このグラフィックスの属性情報が消失してしまう。この際、画像データへの展開で消失する、このグラフィックスの属性情報を取得し、メタデータ１８０３として、その生成された画像データ（ＰＤＬデータ１８０２）に付加する。そして、ＭＦＰ１０００において、ＰＤＬデータ１８０２をビットマップデータに展開し、ベクタデータを生成する際に、メタデータ１８０３が参照される。即ち、ベクタライズ処理を行う時に、トラッピング等を用いることにより、その円形のオブジェクトと四角形のオブジェクトが重なり合った領域を別のオブジェクトとして誤認識してしまうことなく、透過グラフィックス部分を正確に認識することができる。

【００８７】

〔実施の形態３〕

本実施の形態３では、プリンタドライバでメタデータに格納したホスト情報（アプリ名、バージョン）を利用したベクタライズの精度向上の例を説明する。

【００８８】

例えば、アプリケーションによっては、アプリケーションで色処理から中間調処理（ＨＴ）まで実施するものがある。そのためＨＴ後の塗り潰しパターンは、プリンタが内蔵しているパターンを用いる一般のアプリケーションとは異なる。しかし、アプリデータを作成したアプリケーションの特徴等をどこまで表現できるかはプリンタに依存することになる。また従来は、プリンタでレンダリングイメージをベクタライズする際にはアプリケーションの特徴等の情報は利用できず、一律な処理を行っていた。

【００８９】

従って、上述のようなアプリケーションでＨＴを実行したデータであっても、プリンタでベクタライズ処理される場合には、そのプリンタが内蔵する特定のパターンを用いてベクタライズされてしまう。

【００９０】

そこで、プリンタドライバ１５０１が、それらの情報を取得し、メタデータとして画像データに付加してプリンタに供給する。これにより、ベクタライズ時にそのメタデータを参照することができ、アプリケーションの特徴に沿ったベクタライズ処理を実行できる。

【００９１】

図１９は、本発明の実施の形態３に係る動作を説明する図である。

【００９２】

例えば、上述のようなアプリケーションがＨＴを実行した部分は、メタデータ１９０１として、アプリデータ１９００に付加してＭＦＰ１０００に送信する。これによりプリンタでは、そのアプリデータのベクタライズ時に、そのメタデータを参照することにより、そのプリンタが内蔵しているパターンではなく、アプリケーションがＨＴを実行したパターンそのものを利用してベクタライズを行うことができる。

【００９３】

尚、この場合、プリンタのＵＩに「アプリの特徴を再現する」ボタンを用意して、アプリケーションがＨＴを実行したパターン、或はプリンタ内蔵のパターンを使用するかを指定できるようにしても良い。

【００９４】

図２０は、本実施の形態に係るプリンタドライバの処理を説明するフローチャートである。なお、この一連の動作の制御は、ＰＣ１５５０のＣＰＵ１５３１がＲＯＭ１５３５に格納されたプログラムに基づいて実行するものとする。

【００９５】

先ずステップＳ２００１で、アプリケーションが作成したアプリデータを受け取る。次にステップＳ２００２に進み、受け取ったアプリデータを展開して画像データを生成する。続くステップＳ２００３では、ユーザからの指示がＭＦＰ１０００に画像データを印刷する処理を依頼するものか、或いは保存する処理を依頼するものかを判定する。この判定の結果、ユーザからの指示がＭＦＰ１０００に画像データを保存する処理を依頼するものである場合に、ステップＳ２００４に進む。ステップＳ２００４では、アプリケーション

10

20

30

40

50

が作成したアプリデータに含まれ、展開後の画像データに含まれていない情報（文字コードの一部、オブジェクトの属性情報、或いはアプリデータを作成したアプリケーションに関する情報等）を取得する。この処理は、図15のアプリ情報取得部1510、ホスト情報取得部1512、描画命令取得部1511、メタデータ生成部1513により実行される。そしてステップS2005で、取得した情報に基づいてメタデータを生成する。次にステップS2006に進み、ステップS2002で生成された画像データに、ステップS2005で生成したメタデータを付加する。そしてステップS2007に進み、メタデータが付加された画像データを外部機器（MF P 1000）に出力する。なお、ステップS2003の判定の結果、ユーザからの指示がMF P 1000に画像データを印刷する処理を依頼するものである場合は、ステップS2004乃至S2006の処理は省略してステップS2007に進む。即ち、この場合は、メタデータが付加されていない画像データが外部装置に送信される。

10

【0096】

なお、以上の説明では、ホストコンピュータ側のプリンタドライバがアプリデータを画像データに展開した状態で、PDLデータとしてMF Pに送信する例について説明した。しかしながら、ホストコンピュータからMF Pに画像データを送信する場合に、PDLデータ以外のデータ形式で送信するようにしても構わない。つまり、例えば、上記の構成をホストコンピュータ側で全てのアプリデータを画像データ（例えばビットマップデータ）に展開した状態でMF P側に送信するホストベース型の印刷システムで採用してもよい。即ち、アプリデータをホストコンピュータ側で画像データに展開することによりアプリデータに含まれている一部の情報が消失するような場合であれば、上記の構成を適用することが可能である。また更に、上記の説明では、アプリデータを画像データに展開することにより消失する情報を取得して付加情報として画像データに付加する構成をホストコンピュータ側のプリンタドライバが備えている例について説明したが、他の態様であっても構わない。即ち、上記プリンタドライバが備える各構成を、MF Pが備えるようにすることもできる。つまり、ホストコンピュータ或いはMF Pが備えるアプリケーションで作成されたアプリデータを、MF Pにおいて画像データに展開して出力する際に、この展開処理により消失する情報を画像データの付加情報として付加するようにしても構わない。

20

【0097】

（他の実施形態）

30

以上、本発明の実施形態について詳述したが、本発明は、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。

【0098】

なお、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を、システム或いは装置に直接或いは遠隔から供給し、そのシステム或いは装置のコンピュータが該供給されたプログラムを読み出して実行することによっても達成され得る。その場合、プログラムの機能を有していれば、形態は、プログラムである必要はない。

【0099】

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、本発明のクレームでは、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。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、プログラムの形態を問わない。

40

【0100】

プログラムを供給するための記録媒体としては、様々なものを使用できる。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）などである。

【0101】

50

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。その場合、ダウンロードされるのは、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であってもよい。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、本発明のクレームに含まれるものである。

【0102】

10

また、本発明のプログラムを暗号化してCD-ROM等のコンピュータ読み取り可能な記憶媒体に格納してユーザに配布する形態としても良い。その場合、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が実行可能な形式でコンピュータにインストールされるようにする。

【0103】

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形態以外の形態でも実現可能である。例えば、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。

20

【0104】

更に、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るようにしてもよい。この場合、その後で、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】

【0105】

【図1】本実施の形態に係るMFPの構成を示す図である。

【図2】本実施の形態に係るMFPのMFP制御部の構成例を説明するブロック図である

30

【図3】本実施の形態に係る画像処理システムの全体構成を示すブロック図である。

【図4】本実施の形態に係るMFPの動作を制御するMFP制御部のソフトウェアの構成を示す機能ブロック図である。

【図5】本実施の形態に係るMFPがコピー動作を実行するときのデータフローを説明する図である。

【図6】図5で示したメタデータ生成処理の具体的な処理の流れを示す図である。

【図7】PDLプリント時のデータ処理の流れを示す図である。

【図8】入力画像に対して領域分割を行った場合の一例を示す図である。

【図9】本実施の形態に係るMFP制御部によるドキュメントの生成処理を示すフローチャートである。

40

【図10】本実施の形態に係るMFP制御部によるドキュメントの印刷処理を示すフローチャートである。

【図11】本実施の形態に係るMFP制御部によるPDLからドキュメントを生成して印刷する処理を示すフローチャートである。

【図12】本実施の形態に係るドキュメントのデータ構造を示す図である。

【図13】ドキュメントデータの具体例を示す図である。

【図14】図8で説明したデータ構造が、メモリ或はファイルにどのように配置されるのかを示す図である。

【図15】本実施の形態に係るPCからプリンタに印刷データを送信して印刷させる印刷

50

システムにおけるプリンタドライバの機能構成を説明する図である。

【図 1 6】本発明の実施の形態に係るプリンタドライバが生成したメタデータを用いて、PDLからのドキュメントを生成して印刷する処理の流れを説明する図である。

【図 1 7】本実施の形態に係るプリンタドライバによるメタデータの生成処理を説明する図である。

【図 1 8】本発明の実施の形態 2 を説明する図である。

【図 1 9】本発明の実施の形態 3 に係る動作を説明する図である。

【図 2 0】本実施の形態に係るプリンタドライバの処理を説明するフローチャートである。

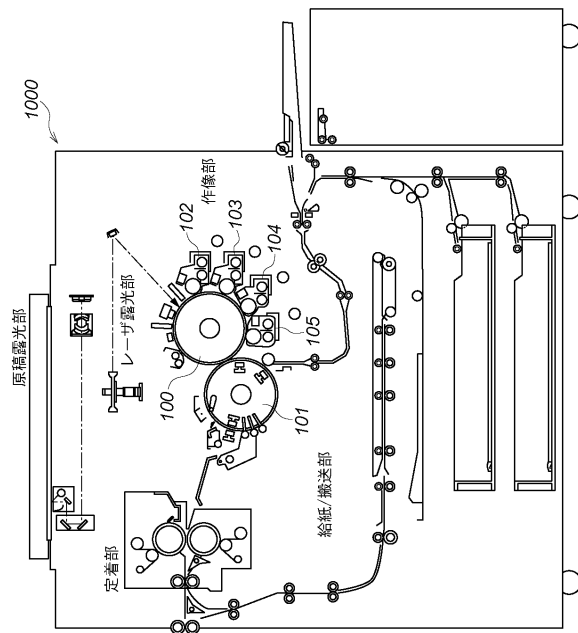
【符号の説明】

【 0 1 0 6 】

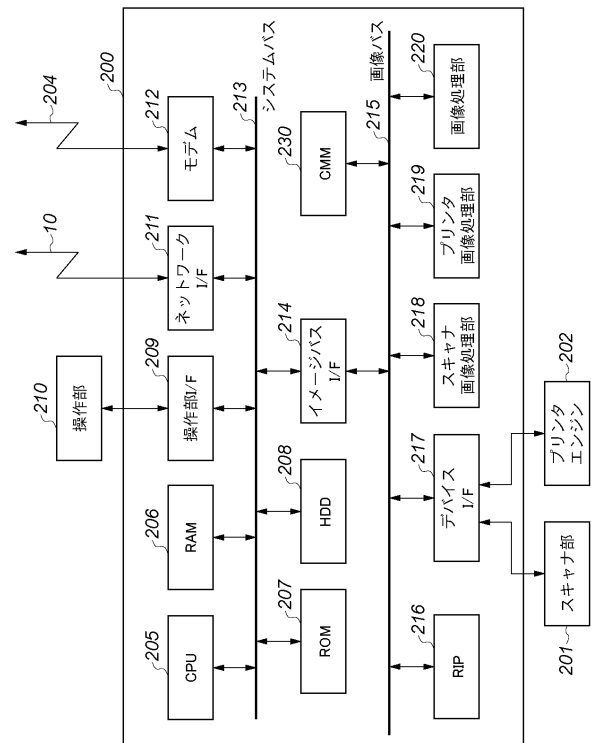
- 1 5 1 0 アプリ情報取得部
- 1 5 1 1 描画命令取得部
- 1 5 1 2 ホスト情報取得部
- 1 5 1 3 メタデータ生成部

10

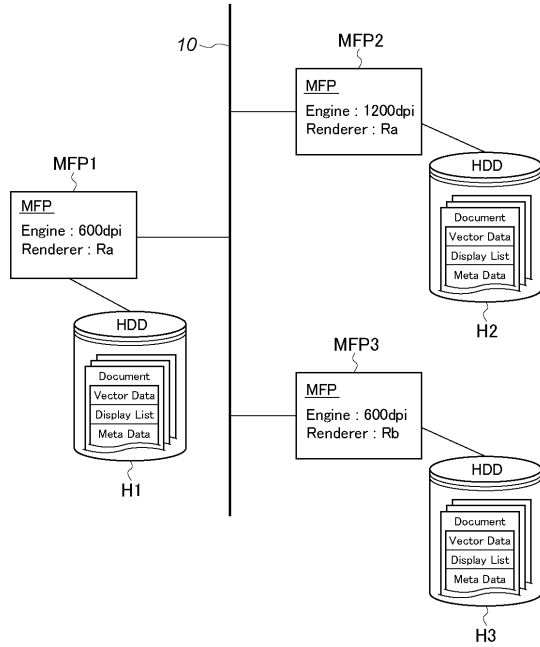
【図 1】



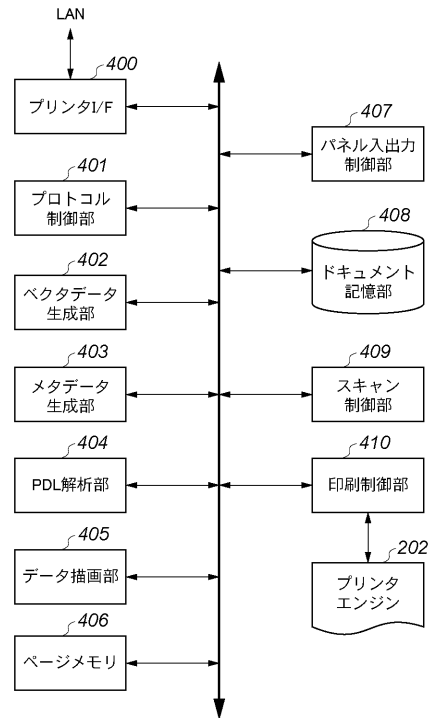
【図 2】



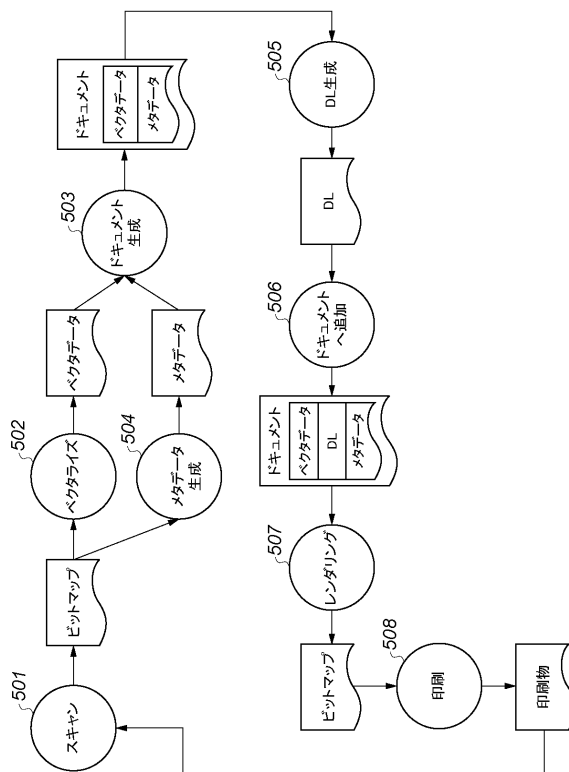
【図 3】



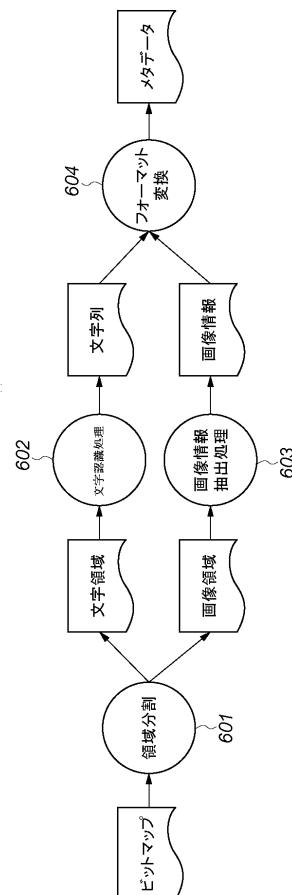
【図 4】



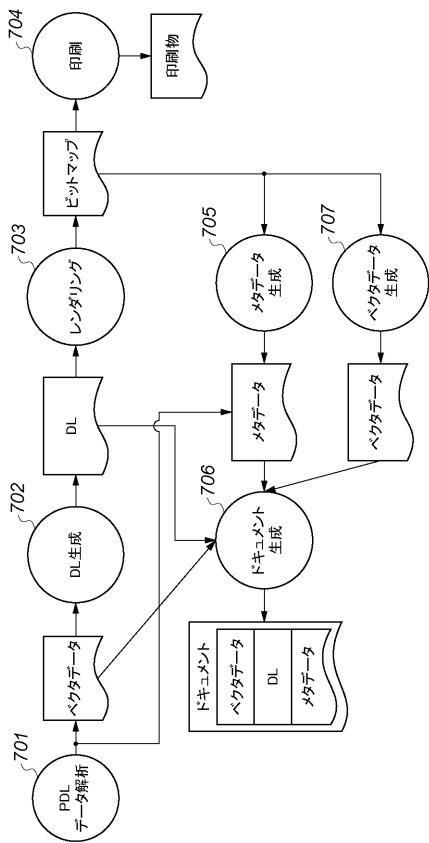
【図 5】



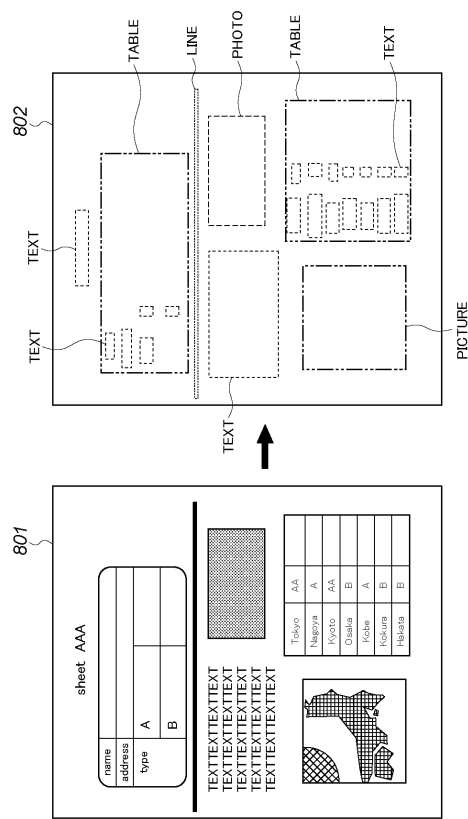
【図 6】



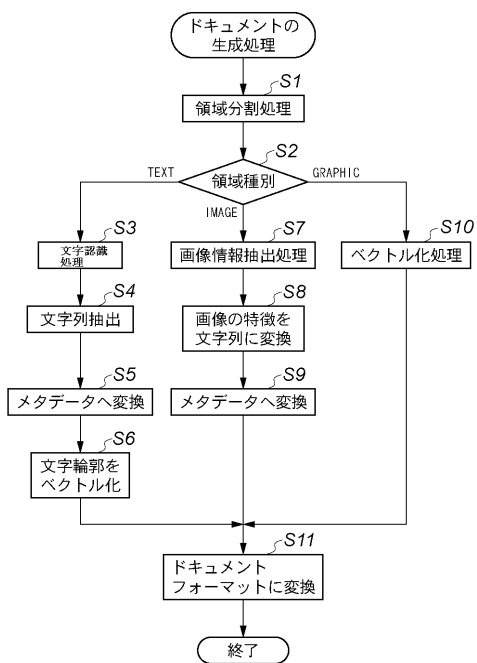
【図 7】



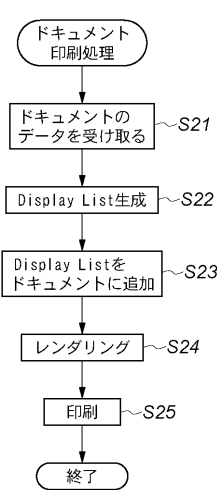
【図 8】



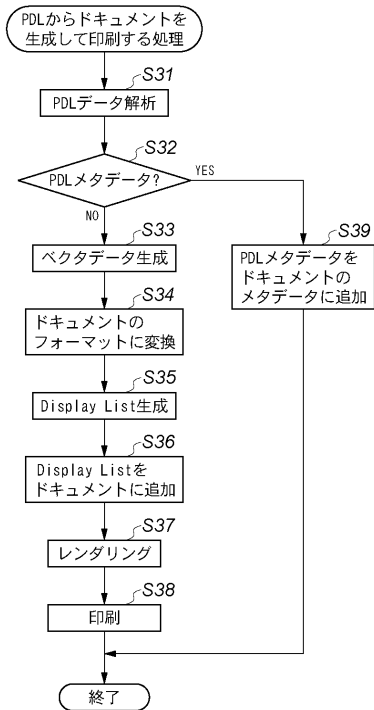
【図 9】



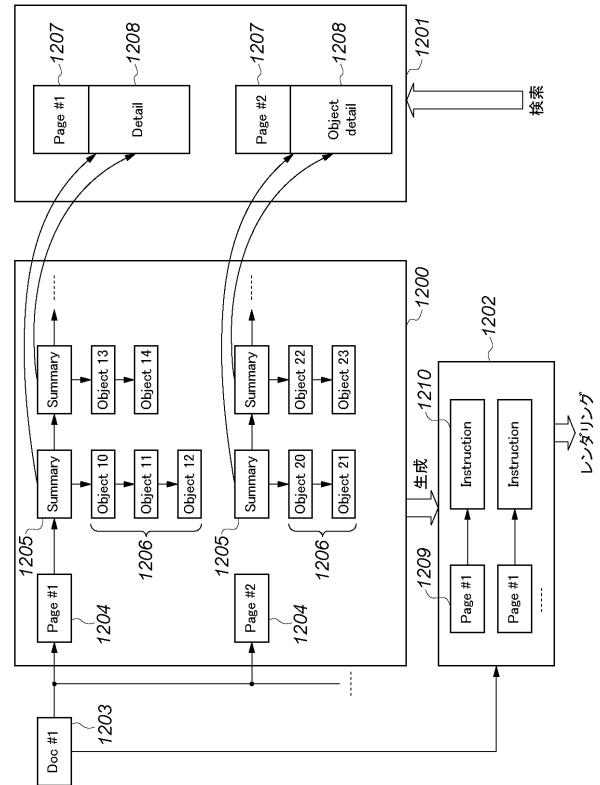
【図 10】



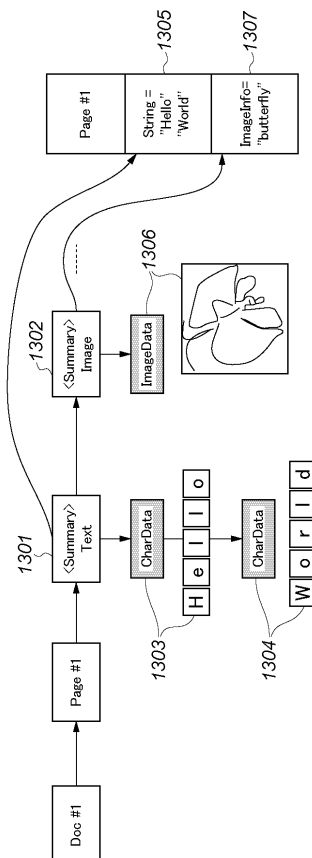
【図 1 1】



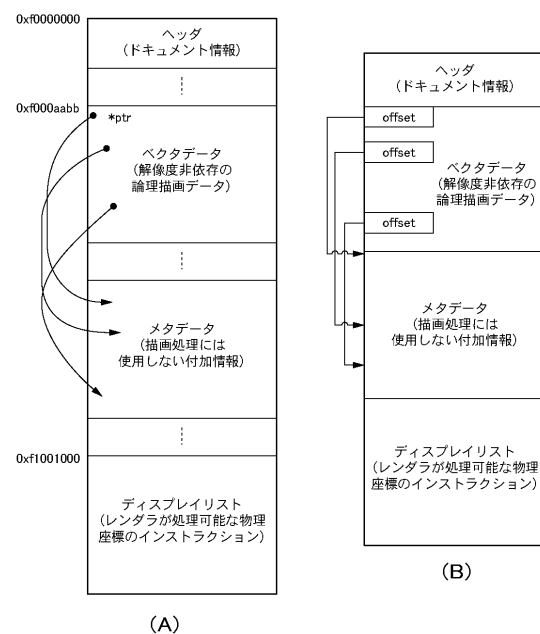
【図 1 2】



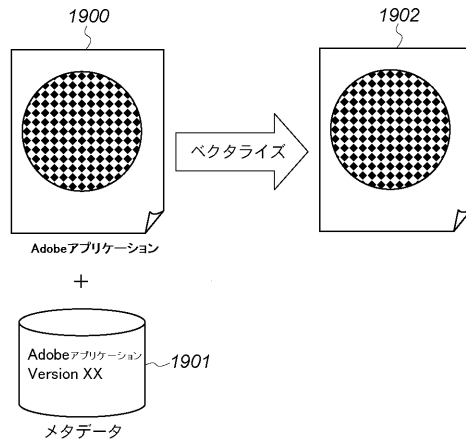
【図 1 3】



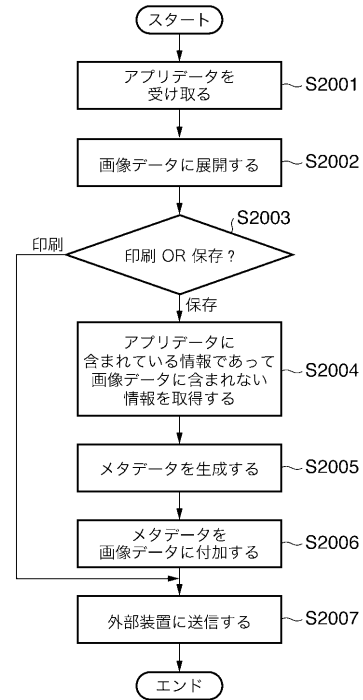
【図 1 4】



【図 19】



【図 20】



フロントページの続き

(72)発明者 大石 哲
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 安島 智也

(56)参考文献 特開平10-269036(JP,A)
特開平11-129556(JP,A)
特開2000-134468(JP,A)
特開2001-256007(JP,A)
特開2005-108199(JP,A)
特開2006-137148(JP,A)
特開2008-020946(JP,A)

(58)調査した分野(Int.Cl., DB名)
G06F 3/12