

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号  
特許第5111268号  
(P5111268)

(45) 発行日 平成25年1月9日(2013.1.9)

(24) 登録日 平成24年10月19日(2012.10.19)

(51) Int.Cl.

F I

G O 6 T 1/00 (2006.01)

G O 6 F 17/30 (2006.01)

G O 6 T 1/00 2 O O D

G O 6 F 17/30 2 1 O A

G O 6 F 17/30 3 1 O C

請求項の数 10 (全 30 頁)

(21) 出願番号	特願2008-179204 (P2008-179204)	(73) 特許権者	000001007
(22) 出願日	平成20年7月9日(2008.7.9)		キヤノン株式会社
(65) 公開番号	特開2010-20468 (P2010-20468A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成22年1月28日(2010.1.28)	(74) 代理人	110001243
審査請求日	平成23年7月5日(2011.7.5)		特許業務法人 谷・阿部特許事務所
		(74) 代理人	100077481
			弁理士 谷 義一
		(74) 代理人	100088915
			弁理士 阿部 和夫
		(72) 発明者	波潟 健
			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	坂上 努
			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、画像処理方法、そのプログラムおよび記憶媒体

(57) 【特許請求の範囲】

【請求項1】

P D Lデータをレンダリングすることにより画像データを得るレンダリング手段と、  
前記レンダリング手段で得た画像データの中から文字オブジェクトを抽出する抽出手段と、

前記抽出手段により抽出された文字オブジェクトに対して文字認識処理を実行することにより文字コード情報を得る文字認識手段と、

前記P D Lデータに含まれる第2の文字コード情報を抽出する文字コード抽出手段と、  
前記文字認識手段により得られた前記文字コード情報と前記文字コード抽出手段によって抽出された前記第2の文字コード情報とを比較し、当該比較結果に従って、より信頼性の高い文字コード情報を採用する統合手段と、

前記統合手段によって採用された前記より信頼性の高い文字コード情報を含むメタデータを前記画像データに付与するメタデータ付与手段と

を備えることを特徴とする画像処理装置。

【請求項2】

前記レンダリング手段は、前記P D Lデータに対する通常のレンダリング処理によりレンダリングした画像データを第1のレンダリングバッファに保存し、前記P D Lデータに含まれる文字オブジェクトを変換してレンダリングした画像データを第2のレンダリングバッファに保存し、

前記文字認識手段は、前記第2のレンダリングバッファに保存された画像データに対し

て文字認識処理を実行することにより文字コード情報を得て、

前記メタデータ付与手段は、前記メタデータを前記第1のレンダリングバッファに保存された画像データに付与する

ことを特徴とする請求項1に記載の画像処理装置。

【請求項3】

前記レンダリング手段で行われる前記PDLデータに含まれる文字オブジェクトの変換処理は、当該PDLデータに含まれる文字オブジェクトの座標、サイズ、フォント情報、修飾情報のうちの少なくともいずれかを変更する処理であ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】

前記文字コード情報と前記第2の文字コード情報との比較は、各文字コード情報に対して形態素解析を行った結果を比較し、その結果、不明な語がより少ない方の文字コード情報を前記メタデータとして採用することを特徴とする請求項1に記載の画像処理装置。

【請求項5】

前記メタデータ付与手段は、前記メタデータを、前記画像データ内のオブジェクトに付与することを特徴とする請求項1に記載の画像処理装置。

【請求項6】

前記画像データ内のオブジェクトをベクトルデータに変換するベクトル化手段を更に備え、

前記メタデータ付与手段は、前記メタデータを、前記ベクトル化手段で変換された前記画像データ内のオブジェクトのベクトルデータに付与する

ことを特徴とする請求項1に記載の画像処理装置。

【請求項7】

請求項1に記載の画像処理装置と、アプリケーションから描画情報を受け取って該描画情報に基づいて変換したPDLデータを前記画像処理装置に対して送信する機能を持つプリンタドライバを含むホストコンピュータとがネットワークを介して接続した画像処理システムであって、

前記プリンタドライバは、前記アプリケーションから受け取った描画情報をPDLデータに変換する際に、当該PDLデータに含まれる文字オブジェクトの座標、サイズ、フォント情報、修飾情報のうちの少なくともいずれかを変更する変換手段を備える

ことを特徴とする画像処理システム。

【請求項8】

PDLデータをレンダリングすることにより画像データを得るレンダリングステップと、

前記レンダリングステップで得た画像データの中から文字オブジェクトを抽出する抽出ステップと、

前記抽出ステップで抽出された文字オブジェクトに対して文字認識処理を実行することにより文字コード情報を得る文字認識ステップと、

前記PDLデータに含まれる第2の文字コード情報を抽出する文字コード抽出ステップと、

前記文字認識ステップにより得られた前記文字コード情報と前記文字コード抽出ステップによって抽出された前記第2の文字コード情報とを比較し、当該比較結果に従って、より信頼性の高い文字コード情報を採用する統合ステップと、

前記統合ステップによって採用された前記より信頼性の高い文字コード情報を含むメタデータを前記画像データに付与するメタデータ付与ステップと

を含むことを特徴とする画像処理方法。

【請求項9】

コンピュータに、請求項8に記載の方法を実行させるためのプログラム。

【請求項10】

コンピュータに、請求項8に記載の方法を実行させるためのプログラムを格納した、コ

10

20

30

40

50

ンピュータが読み取り可能な記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、画像処理装置に関し、特に、画像データを描画要素（オブジェクト）に分割し、分割された描画要素からテキスト情報を中心としたメタデータを抽出して、これに対応する描画要素と関連付けて格納する画像処理装置に関する。

【背景技術】

【0002】

近年、デジタル複合機（Multi Function Peripheral：以下、MFPと略す）は、大容量のHDDを搭載可能になってきている。従来より、MFPには、コピー機能、PDLプリント機能、FAX機能、スキャン画像送信機能などの機能を備えているが、大容量のHDDを搭載することで、これらの機能に加えて、いわゆるBOX機能を提供できるようになってきている。BOX機能とは、原稿をスキャンして得られた画像データやプリントのためにレンダリングされた画像データを、MFP内部のHDDに格納し、あたかも画像ファイリング装置のように使用可能な機能である。

【0003】

ユーザーは、このBOX機能を用いて格納した画像データを、送信したり、プリントしたり、また、他の画像と合成して出力するといった様々な操作を行うことが可能となる。この際、ユーザーの利便性のためには、目的とする画像データを効率よく検索できることが必要である。このため、画像データに対して、画像データ中から抽出したテキスト情報を検索のための検索インデックスとし、検索性の向上に役立てる技術が開発されている。

【0004】

上記の先行技術は、主にスキャンされた画像データを対象としている。その処理フローとしては、まず、原稿をスキャンすることによって得られた画像データから、文字ブロックと考えられる部分を文字領域として切り出す。こうして得られた文字領域内の画像データに対して、OCR（Optical Character Recognition）処理を行う。このOCR処理（文字認識処理）によって得られた文字コードが検索のためのテキスト情報として、元の画像とともに保存される。

【0005】

一方、コンピュータ上のアプリケーションで作成されたドキュメントをファイリング装置に保存するような先行技術がある。特許文献1では、ドキュメントをプリントドライバを介してPDLデータに変換し、PDLデータ中のテキストオブジェクトから文字コードを取り出して、レンダリングされた画像データとともに、テキスト情報として保存する技術が開示されている。

【0006】

【特許文献1】特開平8-147446号公報

【発明の開示】

【発明が解決しようとする課題】

【0007】

一方、アプリケーションの種類によっては、テキストオブジェクトを文字コードで出力しないものなども存在する。すなわち、アプリケーション上でユーザーが文字として描画したものが、PDLデータに変換された際にテキストオブジェクトにならない例や、文字コードを取り出せても検索インデックスとして使用しがたいテキストオブジェクトになる例が存在する。このようなケースでは、PDLデータ中のテキストオブジェクトから文字コードを抽出するだけでは、有効な検索インデックスを付与することができない。

【0008】

以下に、そのようなケースを例示する。

1．大きいサイズのフォントの文字は、アプリケーション、ドライバ、あるいはPDL上で、線分の集まりであるPathデータとして表現され扱われる場合がある。

２．プリンタ側に指定のフォントが無い場合やグラデーションなどを多用した修飾文字の場合は、アプリケーションやドライバ上で文字がイメージデータとして扱われＰＤＬデータとされる場合がある。

３．文字コードだけでは視覚化されるものが何か分からない場合がある（フォントの種類によって視覚化されるものが異なる）。

４．元のテキストの文字列が一文字一文字に分割されて描画コマンドになっている場合には、１文字ずつ別々のテキストデータとして得られてしまい、有意なテキストデータが得られない。

#### 【０００９】

さらに言えば、上記のような問題が生じやすい飾り文字や大サイズの文字は、重要な意味をもつ場合が多く、これらのテキストの文字情報が失われるのは有効なメタデータ付与を妨げる要因となる。

#### 【課題を解決するための手段】

#### 【００１０】

上記課題を解決するために、本発明に係る画像処理装置は、ＰＤＬデータをレンダリングすることにより画像データを得るレンダリング手段と、前記レンダリング手段で得た画像データの中から文字オブジェクトを抽出する抽出手段と、前記抽出手段により抽出された文字オブジェクトに対して文字認識処理を実行することにより文字コード情報を得る文字認識手段と、前記ＰＤＬデータに含まれる第２の文字コード情報を抽出する文字コード抽出手段と、前記文字認識手段により得られた前記文字コード情報と前記文字コード抽出手段によって抽出された前記第２の文字コード情報とを比較し、当該比較結果に従って、より信頼性の高い文字コード情報を採用する統合手段と、前記統合手段によって採用された前記より信頼性の高い文字コード情報を含むメタデータを前記画像データに付与するメタデータ付与手段とを備えることを特徴とする。

#### 【発明の効果】

#### 【００１１】

本発明に拠れば、ＰＤＬデータ中のテキストオブジェクトから文字コードを抽出するよりも信頼性の高い文字コード列を抽出でき、より有効な検索インデックスを付与した文書保存を行うことができる。また、スキャン画像からの検索インデックス付き文書フォーマットによる文書保存との統一を図りつつ、ＰＤＬデータの文書保存が可能である。

#### 【発明を実施するための最良の形態】

#### 【００１２】

##### < 第１の実施形態 >

本発明による画像処理装置およびシステムの好適な実施の形態について、以下、図面に基づいて説明する。

#### 【００１３】

##### [ 画像処理システム ]

はじめに、本発明の一実施形態に係る画像処理システムについて、図１を用いて説明する。図１は、当該画像処理システムの概略構成を示すブロック図である。

#### 【００１４】

本実施形態の画像処理システムは、ＬＡＮ１０１上にＭＦＰ１００、および単数または複数のクライアントＰＣ１０２ - １～ｎ（ホストコンピュータ）を接続した形態をとる。ＭＦＰ１００とクライアントＰＣ１０２に加えて、本システムは文書管理サーバ１０３およびこのサーバに接続されて使用されるデータベース１０４も備えている。

#### 【００１５】

本実施形態において、ＭＦＰ１００は紙の原稿をスキャンすることによってその原稿の画像データを生成する。また、クライアントＰＣ１０２が生成するＰＤＬデータを受信して、ＰＤＬデータの解釈やレンダリングを行う。また、画像データを、画像記録装置（プリンタ）１１３によって紙媒体上に記録したり、ネットワーク上に送信したり、あるいは、ＭＦＰ１００自身が持つ二次記憶装置からなる記憶部１１５に画像データを格納・記憶

10

20

30

40

50

したりする。画像データを格納する手段は、ネットワーク上のデータベース１０４であっても構わない。その際には、ＭＦＰ１００は文書管理サーバ１０３と通信を行い、データベース１０４に画像データが格納されるように適宜制御する。

【００１６】

[ ＭＦＰ（画像処理装置） ]

続いて、本発明の一実施形態であるＭＦＰ１００の構成について、図２を用いて説明する。

【００１７】

図２は、ＭＦＰ１００の構成を示すブロック図である。

【００１８】

ＭＦＰ１００は、図示しないＡｕｔｏ Ｄｏｃｕｍｅｎｔ Ｆｅｅｄｅｒ（以下、ＡＤＦ）を有する画像読み取り装置１１０を備える。この画像読み取り部装置１１０は束状の或いは１枚の原稿の画像を光源で照射し、反射画像をレンズを介して固体撮像素子上に結像する。固体撮像素子は所定解像度（例えば６００ｄｐｉ）および所定輝度レベル（例えば８ビット階調）の画像読み取り信号を生成し、この画像読み取り信号から、ラスタデータよりなる画像データが構成される。画像読み取り装置１１０で読み取られた画像データは、読み取り画像処理部１１１によってＭＦＰ１００内部で処理しやすい画像データに補正処理される。具体的には、読み取り画像処理部１１１は、ノイズ除去処理、ガンマ補正処理、色再現補正処理、フィルタ処理などを行う。

【００１９】

ＭＦＰ１００は、記憶部１１５および画像記録装置１１３を有し、通常の複写機能を実行する際には、読み取り画像処理部１１１で処理された画像データに対し、さらに出力画像処理部１１２によって画像記録装置用の画像処理を行って、記録信号に変換する。複数枚複写する場合には、１頁分の記録信号を一旦記憶部１１５に記憶保持した後、画像記録装置１１３に順次出力して、記録紙上に記録画像を形成する。

【００２０】

ＭＦＰ１００は、ＬＡＮ１０１との接続のためのネットワークＩ／Ｆ１１４を有し、クライアントＰＣ１０２からプリンタドライバを利用して出力するＰＤＬデータを、画像記録装置１１３によって記録する機能を持つ。クライアントＰＣ１０２からプリンタドライバを経由して出力されるＰＤＬデータは、ＬＡＮ１０１からネットワークＩ／Ｆ１１４を経て記憶部１１５の一部に保存される。保存されたＰＤＬデータは取り出されて、ＲＯＭ１２０あるいは記憶部１１５からＲＡＭ１１９にロードされたプログラムをＣＰＵ１１８が実行することによって実現される後述のＰＤＬ処理およびレンダリング処理によって、Ｂｉｔｍａｐデータとして展開される。そして、このＢｉｔｍａｐデータに対し出力画像処理部１１２で、画像記録装置用の画像処理を行って記録可能な記録信号に変換する。この記録信号は、画像記録装置１１３に送出されて、記録紙上に記録画像として記録される。

【００２１】

また、記憶部１１５は、画像読み取り装置１１０からのデータやクライアントＰＣ１０２からプリンタドライバを経由して出力される、ＰＤＬデータをレンダリングしたデータを保存できる機能（以下、この機能をＢＯＸ機能と称す）を有している。

【００２２】

ＭＦＰ１００は、ＭＦＰ１００を操作するためのユーザーインターフェースとして、入力装置１１６と表示装置１１７を備えている。表示装置１１７上には、ＭＦＰ１００の状態を表す情報やＢＯＸ機能により保存した画像データが表示される。また、表示装置１１７はタッチパネルを備え、表示装置１１７上に表示されたボタンをユーザーが触れることでユーザーの操作による指定入力を行うことができる。このほかのユーザー入力を受け付ける装置として、入力装置１１６は数値の入力を行うためのキーを備えている。

【００２３】

ＭＦＰ１００の上記構成要素は、ＣＰＵ１１８で動作するＲＡＭ１１９上のプログラム

10

20

30

40

50

によって、内部バス 1 2 1 を介して制御される。

【 0 0 2 4 】

[ M F P の各機能の動作 ]

次に、本実施形態における M F P の持つ機能について、図 3 ~ 6 を用いて説明する。図 3 は、M F P 1 0 0 のコピー機能・スキャン送信機能・B O X 機能を実現する各処理ブロックによる処理の流れを示す図である。図 4 は、P D L データに対する P D L 処理機能・B O X 機能について各処理ブロックによる処理の流れを示している。これらの機能は、一例として図 5 に示す U I 画面 5 0 0 からのユーザー指示によって処理が開始される。

【 0 0 2 5 】

図 5 において、上部に位置するコピータブ 5 0 4、送信 F A X タブ 5 0 3、B O X タブ 5 0 1 といった各機能別のタブによって、M F P 1 0 0 にどの機能を働かせるかをユーザーに選択させる。図面上では、B O X タブ 5 0 1 が選択されているので、この時に読み込みスタートボタン 5 0 2 を押下すると、M F P 1 0 0 は B O X 格納を行うように制御する。他のタブにも 5 0 2 と同様にその機能を働かせるスタートボタンがあり、B O X 機能の開始の場合と同様に、ボタン押下で該当する機能の処理をスタートするようにする。

【 0 0 2 6 】

M F P 1 0 0 は、コピー機能・スキャン送信機能・B O X 機能のいずれの場合においても、画像読み取り装置 1 1 0 で紙の原稿を読み取り、画像データを得るように制御する。その次に読み取り画像処理部 1 1 1 でスキャン画像用の画像処理を行い、記憶部 1 1 5 の一部を用いて実現する一時的な画像データ保存領域である B i t m a p スプール ( レンダリングバッファ ) 2 0 1 に B i t m a p データを保存する。

【 0 0 2 7 】

コピー機能使用時は、B i t m a p スプール 2 0 1 から取り出した B i t a m p データを出力画像処理部 1 1 2 で画像処理を行って記録信号に変換してから、画像記録装置 1 1 3 に出力するように制御する。

【 0 0 2 8 】

スキャン送信機能においては、C P U 1 1 8 上で動作するプログラムによって実現される画像送信処理部 2 0 2 により、ネットワーク I / F 1 1 4 を介して、U I 上で指定された I P アドレスを持つ P C 等の機器に対して、画像データを送るように制御する。

【 0 0 2 9 】

最後に B O X 機能においては、B i t m a p スプール 2 0 1 から取り出した B i t m a p データを、C P U 上で動作するプログラムによって実現される画像格納処理部 2 0 0 で処理したあと、記憶部 1 1 5 に記憶するようにする。ここで述べた画像格納処理については、後ほど詳細に説明を加える。

【 0 0 3 0 】

M F P 1 0 0 が持つ P D L 処理機能は、ユーザーがクライアント P C 上のアプリケーション 3 0 1 でプリント指示を行うところから処理が始まる。図 4 に示すように、クライアント P C 1 0 2 上で動作するアプリケーション 3 0 1 から、プリンタドライバ 3 0 2 にアプリケーションデータが描画情報として渡される。このデータは、プリンタドライバ 3 0 2 で P o s t S c r i p t や L I P S などの P D L 言語に従うデータ ( P D L データ ) に変換されて、クライアント P C 1 0 2 のネットワーク I / F ( 不図示 ) を介してネットワーク送信される。

【 0 0 3 1 】

M F P 1 0 0 はネットワーク I / F 1 1 4 で受信した P D L データを、P D L データの一時保存領域である P D L データスプール 3 0 3 に保存する。いったんスプールされた P D L データは取り出された後に、P D L 処理部 3 0 4 によって言語解釈されて、レンダラ 3 0 5 が処理できる中間的なデータ構造である D i s p l a y L i s t を生成する。レンダラ 3 0 5 は生成された D i s p l a y L i s t を B i t m a p データに展開して、B i t m a p スプール 2 0 1 に一時保存する。

【 0 0 3 2 】

10

20

30

40

50

MFP100は、プリンタドライバ302の指示が印刷であれば、Bitmapスプール201から画像データを取り出した後に、出力画像処理部112による画像処理を行い、画像記録装置113に記録信号を送出して処理を終える。プリンタドライバの指示が画像データの保存である場合には、後述の画像格納処理200を行った後に、記憶部115に画像データを格納するようにする。

【0033】

プリンタドライバの指示については、図6に示すように、保存を指定するUI項目602と印刷を指定するUI項目601によって、保存指示か、印刷指示かをユーザーに選択させる。

【0034】

10

[画像格納処理]

次に、画像格納処理について図7を用いて説明する。

【0035】

図7は、画像格納処理部200の構成を示すブロック図である。

【0036】

MFP100におけるUI500あるいはドライバUI600によって、文書の保存が指示された場合に、Bitmapスプール201から取り出した画像データをBOXに保存するために行う画像格納処理部の処理の流れの一例を図7に示す。

【0037】

MFP100は、Bitmapスプール201から取り出した画像データを、まず、オブジェクト分割処理部701によって、オブジェクトを含む領域毎に領域分割することによりオブジェクト分割を行う。

20

【0038】

オブジェクト分割後のオブジェクトの種類は、文字、写真、グラフィック（図面、線画、表）、背景、である。このオブジェクト分割処理においては、ビットマップデータのままで、オブジェクト毎にそれを含むように画像領域分割を行い、分割された各々の画像領域に対して、周知の手法でオブジェクトの種類（文字、写真、グラフィック、背景）を判別する。

【0039】

判別対象の画像領域が写真オブジェクトと判別された場合、この領域は、JPEG圧縮処理部702において、JPEG圧縮される。また、背景の場合も同様に、JPEG圧縮される。一方、オブジェクト判別の結果が、グラフィックの場合、ベクトル化処理部703において、ベクトル化処理され、パス化されたデータに変換される。また、オブジェクト判別の結果が、文字の場合には、ベクトル化処理部703において、グラフィックと同様にベクトル化処理され、パス化されたデータに変換される。それとともに、文字の場合には、OCR処理部704にも当該分割領域のBitmapデータが送られ、そのデータに対しOCR処理が施され、文字コード化されたデータが抽出される。これらの全てのオブジェクトデータと、文字コード化されたデータが一つのファイルとしてまとめられる。

30

【0040】

最後に、メタデータ付与処理部705において、各オブジェクトに対して、生成されたメタデータが付与される。このメタデータが付与された各々のオブジェクトデータは、MFP100に内蔵されているBOX115に保存される。ここで保存されたデータは、UI画面に表示するようにしてもよい。

40

【0041】

以下では、メタデータ付与処理、オブジェクト分割処理、ベクトル化処理について、さらに詳細に説明する。

【0042】

[メタデータ付与処理]

まず、メタデータ付与処理について、図8を用いて説明する。

【0043】

50

図 8 は、メタデータ付与処理部 705 で行うメタデータ付与処理のフローチャートである。

【0044】

まず、ステップ S801 において、メタデータを付与するオブジェクトの周囲で一番近くに存在する文字オブジェクトを選択する。

【0045】

次に、ステップ S802 において、ステップ S801 で選択された文字オブジェクトに対して、形態素解析を行う。

【0046】

次いで、ステップ S803 において、ステップ S802 における形態素解析により抽出された単語をメタデータとして、これを対象のオブジェクトに付加する。なお、メタデータ付与対象のオブジェクトが文字オブジェクトの場合もあり、このときの、周囲で一番近くにあるオブジェクトは、この文字オブジェクト自身をも含む。すなわち、この場合、文字オブジェクトから抽出した単語がその文字オブジェクトのメタデータとして付与されることとなる。

【0047】

また、メタデータの作成には、形態素解析によって抽出した単語だけではなく、そのオブジェクトから抽出した色ヒストグラムなどの画像特徴量や、形態素解析だけでなく、より高度な構文解析等により得られる文章要約情報なども含めることが可能である。

【0048】

このようにして得られたメタデータは、オブジェクトの各々に対して記述されて、オブジェクトの画像データ（ベクトル化または J P E G 圧縮されたデータ）とともに、B O X に保存される。

【0049】

[ベクトル化結果のデータフォーマット]

次に、ベクトル化処理部 703 でベクトル化されたデータのフォーマットについて、図 9 を用いて説明する。

【0050】

図 9 は、ベクトル化処理部 703 でベクトル化されたデータのフォーマットの一例を示す図である。本実施形態では、ベクトル化処理部 703 でベクトル化されたデータを S V G 形式で表記しているが、これに限定されるものではない。

【0051】

図 9 では説明のために、オブジェクトの表記を枠で囲っている。枠：901 はイメージ属性を示し、そこには、イメージオブジェクトの領域の示す領域情報とビットマップ情報が示されている。枠：902 はテキストオブジェクトの情報が示されており、枠：903 では、枠：902 で示した内容、すなわちテキストオブジェクトの情報をベクターオブジェクトとして表現している。枠：904 は、表オブジェクトなどのラインアートを表す。

【0052】

メタデータ付与処理部 705 で付与されるメタデータは、例えば、それが“山田 太郎”である場合、各ブロックの末尾に、以下の、

```
<metadata>
  <text_index>山田 太郎</text_index>
</metadata>
```

のように付与すればよい。

【0053】

[オブジェクト分割処理]

次に、オブジェクト分割処理について、図 10、11 を用いて説明する。

【0054】

図 10 は、オブジェクト分割処理による画像データのオブジェクト分割結果の一例を示す図である。また、図 11 は、オブジェクト分割したときの各属性のブロック情報および

10

20

30

40

50



入力ファイル情報の例を示す図である。

【 0 0 5 5 】

オブジェクト分割処理部 7 0 1 は、図 1 0 右側の画像 1 0 0 2 に示すように、入力画像 1 0 0 1 ( 図 1 0 左側 ) を属性ごとに矩形ブロックに分割する。前述のように、本実施形態においては、矩形ブロックの属性としては、文字、写真、グラフィック ( 図面、線画、表など ) とするが、もちろん、これに限られるものではなく、色文字や黒文字、あるいは自然画像や C G 画像などに細分化することも可能である。

【 0 0 5 6 】

オブジェクト分割処理においては、まず、B i t m a p スプール 2 0 1 から取り出されて R A M 上に格納されたイメージデータを白黒に 2 値化し、黒画素からなる輪郭で囲まれる画素塊を抽出する。

10

【 0 0 5 7 】

さらに、このように抽出された画素塊のうち黒画素塊の大きさを評価し、大きさが所定値以上の黒画素塊の内部にある白画素塊に対する輪郭追跡を行う。さらに、白画素塊に対する大きさ評価、およびその内部の黒画素塊の追跡というように、内部の画素塊の大きさが所定値以上である限り、再帰的に内部の画素塊の抽出、および輪郭追跡を行う。なお、画素塊の大きさは、例えば画素塊の面積によって評価される。

【 0 0 5 8 】

このようにして得られた画素塊に外接する矩形ブロックを生成し、矩形ブロックの大きさ、および形状に基づき属性を判定する。

20

【 0 0 5 9 】

例えば、縦横比が 1 に近く、大きさが一定の範囲の矩形ブロックは文字領域矩形ブロックの可能性のある文字相当ブロックとする。そして、近接する文字相当ブロックが規則正しく整列しているときに、これら文字相当ブロックを纏めた新たな矩形ブロックを生成し、この新たな矩形ブロックを文字領域矩形ブロックとする。

【 0 0 6 0 】

また扁平な画素塊、もしくは、一定大きさ以上でかつ四角形の白画素塊を整列した状態で内包する黒画素塊をグラフィック領域矩形ブロック、それ以外の不定形の画素塊を写真領域矩形ブロックとする。

【 0 0 6 1 】

30

オブジェクト分割処理では、このようにして生成された矩形ブロックのそれぞれについて、図 1 1 に示す、属性等のブロック情報および入力ファイル情報を生成する。

【 0 0 6 2 】

図 1 1 において、同図に示すブロック情報には、各ブロックの属性、その位置の座標 X、座標 Y、幅 W、高さ H、O C R 情報が含まれる。属性は 1 ~ 3 の数値で与えられ、ここで 1 は文字領域矩形ブロック、2 は写真領域矩形ブロック、3 はグラフィック領域矩形ブロックを示す。座標 X、座標 Y は入力画像における各矩形ブロックの始点の X、Y 座標 ( 左上角の座標 ) である。幅 W、高さ H は、それぞれ矩形ブロックの X 座標方向の幅、Y 座標方向の高さである。O C R 情報は、各ブロックのメタデータとして使用可能な O C R 結果があるか否かを示す。さらに入力ファイル情報として矩形ブロックの個数を示すブロック総数 N が含まれる。

40

【 0 0 6 3 】

これらの矩形ブロックごとのブロック情報は、特定領域でのベクトル化に利用される。またブロック情報によって、特定領域とその他の領域を合成する際の相対位置関係を特定でき、入力画像のレイアウトを損なわずにベクトル化領域とラスタデータ領域を合成することが可能となる。

【 0 0 6 4 】

[ ベクトル化処理 ]

次に、ベクトル化処理部 7 0 3 の処理について、図 1 2 を用いて説明する。

【 0 0 6 5 】

50

図12は、ベクトル化処理部703における処理の流れを示すフローチャートである。

【0066】

はじめに、ステップS1201で、処理の対象となる特定領域が文字領域矩形ブロックであるか否かを判断する。このとき、特定領域が文字領域矩形ブロックであればステップS1202以下のステップに進み、以下に説明するようにパターンマッチングの一手法を用いて文字認識を行い、対応する文字コードを得る。一方、特定領域が文字領域矩形ブロックでない場合は、ステップS1212に移行する。

【0067】

ステップS1202では、特定領域に対し横書き、縦書きの判定（組み方向判定）をおこなうために、特定領域内で画素値に対する水平・垂直の射影を取る。

【0068】

次いで、ステップS1203で、ステップS1202の射影の分散を評価する。水平射影の分散が大きい場合は横書き、垂直射影の分散が大きい場合は縦書きと判断する。

【0069】

次に、ステップS1204で、ステップS1203での評価結果に基づき、組み方向を判定し、組み方向に応じた行の切り出しを行う。さらに、切出した行から文字を切り出して文字画像を得る。文字列および文字への分解は、横書きならば水平方向の射影を利用して行を切り出し、切り出された行に対する垂直方向の射影を基に、文字を切り出す。縦書きの文字領域に対しては、水平方向と垂直方向について、逆の処理を行う。ここでの行、文字の切り出しに際して、文字のサイズも検出し得る。

【0070】

次に、ステップS1205では、ステップS1204で切り出された各文字について、文字画像から得られる特徴量を数十次元の数値列に変換した特徴ベクトル（ここでは、観測特徴ベクトルと称す）を生成する。この観測特徴ベクトルの抽出には種々の公知手法があり、例えば、文字をメッシュ状に分割し、各メッシュ内の文字線を方向別に線素としてカウントしたメッシュ数次元ベクトルを特徴ベクトルとする方法がある。上記観測特徴ベクトルは、このような周知の手法を用いて生成可能である。

【0071】

次に、ステップS1206で、ステップS1205で得られた観測特徴ベクトルと、あらかじめフォントの種類ごとに求められている辞書特徴ベクトルとを比較し、観測特徴ベクトルと辞書特徴ベクトルとの距離を算出する。

【0072】

次いで、ステップS1207において、ステップS1206で算出された距離を評価し、最も距離の近いフォントの種類を、その文字の認識結果とする。

【0073】

次に、ステップS1208では、ステップS1207における距離評価において、最短距離が所定値よりも大きいかなにより、類似度を判断する。最短距離が所定値以上の場合は、辞書特徴ベクトルにおいて、形状が類似する他の文字に誤認識している可能性が高い。そこで上記最短距離が所定値以上の場合は、ステップS1207の認識結果を採用せず、ステップS1211の処理に進む。一方、上記最短距離が所定値より低い（小さい）ときは、ステップS1207の認識結果を採用し、ステップ1209に進む。

【0074】

ステップS1209（フォント認識ステップ）では、文字フォントの認識処理を行う。本実施形態では、文字認識の際に用いる、フォントの種類数分の辞書特徴ベクトルを、文字形状種すなわちフォント種に対して複数用意しておく。そして、パターンマッチングの際に、文字コードとともにフォント情報としてマッチするフォント種を出力することで、文字フォントを認識し得る。

【0075】

次いでステップS1210では、上記文字認識およびフォント認識によって得られた文字コードおよびフォント情報と、あらかじめ用意された対応するアウトラインデータを用い

10

20

30

40

50

て、各文字をベクトルデータに変換する。なお、入力画像がカラーの場合は、カラー画像から各文字の色を抽出して、得られた色情報をベクトルデータとともに記録する。

【 0 0 7 6 】

ステップ S 1 2 1 1 では、文字を一般的なグラフィックと同様に扱い、該当文字をアウトライン化する。すなわち誤認識を起こす可能性の高い文字については、可視的にイメージデータに忠実なアウトラインのベクトルデータを生成する。

【 0 0 7 7 】

ステップ S 1 2 1 2 では、特定領域が文字領域矩形ブロックでないとき、画像の輪郭に基づいてベクトル化の処理を実行する。

【 0 0 7 8 】

以上の処理により、文字領域矩形ブロックに属するイメージ情報を、形状、大きさ、色についてほぼ忠実なベクトルデータに変換出来る。

【 0 0 7 9 】

[ グラフィック領域のベクトル化 ]

次に、グラフィック領域のベクトル化について、図 1 3、1 4 を用いて説明する。

【 0 0 8 0 】

図 1 3 は、ベクトル化の処理における角抽出の処理を説明するための図であり、図 1 4 は、ベクトル化の処理における輪郭線まとめの処理を説明するための図である。

【 0 0 8 1 】

オブジェクト分割処理において、特定領域が、文字領域矩形ブロック以外の領域、すなわちグラフィック領域矩形ブロックと判断されたときは、特定領域内で抽出された黒画素塊の輪郭をベクトルデータに変換する。

【 0 0 8 2 】

文字領域以外の領域のベクトル化においては、まず線画等を直線および / または曲線の組み合わせとして表現するために、曲線を複数の区間 ( 画素列 ) に区切る「角」を検出する。「角」とは曲率が極大となる点である。例えば、図 1 3 の曲線上の画素  $P_i$  が「角」か否かの判断は以下のように行う。

【 0 0 8 3 】

すなわち、 $P_i$  を起点とし、曲線に沿って  $P_i$  から両方向に所定画素 ( $k$  個とする) ずつ離れた画素  $P_{i-k}$ 、 $P_{i+k}$  を線分  $L$  で結ぶ。画素  $P_{i-k}$ 、 $P_{i+k}$  間の距離を  $d_1$ 、線分  $L$  と画素  $P_i$  との距離を  $d_2$ 、曲線の画素  $P_{i-k}$ 、 $P_{i+k}$  間の弧の長さを  $A$  とする。 $d_2$  が極大となるとき、あるいは比 ( $d_1 / A$ ) が閾値以下となるときに画素  $P_i$  を「角」と判断する。

【 0 0 8 4 】

「角」によって分割された画素列を、直線あるいは曲線で近似する。直線への近似は最小二乗法等により実行し、曲線への近似は 3 次スプライン関数などを用いる。画素列を分割する「角」の画素は近似直線あるいは近似曲線における、始端または終端となる。

【 0 0 8 5 】

さらにベクトル化された輪郭内に白画素塊の内輪郭が存在するか否かを判断し、内輪郭が存在するときはその輪郭をベクトル化し、その内輪郭の内輪郭というように、再帰的に反転画素の内輪郭をベクトル化する。

【 0 0 8 6 】

以上のように、輪郭の区分線近似を用いれば、任意形状の図形のアウトラインをベクトル化することができる。元原稿がカラーの場合は、カラー画像から図形の色を抽出して、得られた色情報をベクトルデータとともに記録する。

【 0 0 8 7 】

図 1 4 に示すように、ある注目区間で外輪郭  $PR_j$  と、内輪郭  $PR_{j+1}$  あるいは別の外輪郭が近接している場合、2 つあるいは複数の輪郭線をひとまとめにし、太さを持った線として表現することができる。例えば、輪郭  $PR_{j+1}$  の各画素  $P_i$  から輪郭  $PR_j$  上で最短距離となる画素  $Q_i$  までの距離  $PQ_i$  を算出し、 $PQ_i$  のばらつきがわずかである

10

20

30

40

50

場合には、注目区間を画素  $P_i$ 、 $Q_i$  の中点  $M_i$  の点列に沿った直線または曲線で近似し得る。近似直線、近似曲線の太さは、例えば距離  $PQ_i$  の平均値とする。線や線の集合体である表罫線は、太さを持つ線の集合とすることにより、効率よくベクトル表現することができる。

【0088】

輪郭線まとめの処理の後、ベクトル化全体の処理を終了する。なお写真領域矩形ブロックについては、ベクトル化せず、イメージデータのままとする。

【0089】

[ベクトルデータのグループ化]

次に、ベクトルデータのグループ化について、図15を用いて説明する。

10

【0090】

図15は、図12を用いて説明したベクトル化処理により生成されたベクトルデータのグループ化の処理を示すフローチャートである。

【0091】

ベクトル化処理により線図形等のアウトラインをベクトル化した後、ベクトル化された区分線を図形オブジェクトごとにグループ化する。

【0092】

はじめに、ステップS1501で、各ベクトルデータの始点、終点を算出する。

【0093】

次いで、ステップS1502（図形要素検出）において、ステップS1201で求められた始点、終点情報を用いて、図形要素を検出する。ここで図形要素とは、区分線が構成している閉図形であり、検出に際しては、始点、終端となっている共通の「角」の画素においてベクトルを連結する。すなわち、閉形状を構成する各ベクトルはその両端にそれぞれ連結するベクトルを有しているという原理を応用する。

20

【0094】

次いで、ステップS1503において、次に図形要素内に存在する他の図形要素、もしくは区分線をグループ化し、一つの図形オブジェクトとする。また、図形要素内に他の図形要素、区分線が存在しない場合はその図形要素自体を図形オブジェクトとする。

【0095】

[図形要素の検出]

30

次に図形要素の検出処理について、図16を用いて説明する。

【0096】

図16は、図15を用いて説明したベクトルデータのグループ化の処理でグループ化されたベクトルデータに対する図形要素検出の処理を示すフローチャートである。

【0097】

前述のステップS1502（図形要素検出）の処理は、図16の各ステップによって実行される。

【0098】

はじめに、ステップS1601において、ベクトルデータから両端に連結していない不要なベクトルを除去し、閉図形を構成するベクトルを抽出する。

40

【0099】

次に、ステップS1602で、閉図形を構成するベクトルについて、いずれかのベクトルの端点（始点または終点）を開始点とし、一定方向、例えば時計回りに、順にベクトルを探索する。すなわち、他端点において他のベクトルの端点を探索し、所定距離内の最近接端点を連結ベクトルの端点とする。閉図形を構成するベクトルを1まわりして開始点に戻ったとき、通過したベクトルを全て一つの図形要素を構成する閉図形としてグループ化する。また、閉図形内部にある閉図形構成ベクトルも全てグループ化する。さらにまだグループ化されていないベクトルの始点を開始点とし、同様の処理を繰り返す。

【0100】

最後に、ステップS1603において、ステップS1601で除去された不要ベクトル

50

のうち、ステップ S 1 6 0 2 で閉図形としてグループ化されたベクトルに端点が近接しているベクトルを検出し、これも一つの図形要素としてグループ化する。

#### 【 0 1 0 1 】

以上の処理によって図形要素を、再利用可能な個別の図形オブジェクトとして扱う事が可能になる。

#### 【 0 1 0 2 】

##### [ 表示方法 ]

M F P 1 0 0 は、B O X に保存した文書を、その確認のため、あるいは、検索などの操作のために、表示装置 1 1 7 により表示し、入力装置 1 1 6 などによって操作することができる。以下、このような操作のための U I 表示に関して図 1 7 ~ 2 1 を用いて詳細な説明を行う。

#### 【 0 1 0 3 】

図 1 7 ~ 2 1 は、それぞれ、本実施形態におけるユーザーインターフェースの一表示形態を示す図である。

#### 【 0 1 0 4 】

図 1 7 は、特に、B O X 内に保存されているデータの一覧 ( B O X 文章一覧 1 7 0 1 ) を U I 上に表示したものを示している。同図 1 7 0 2 に示すように一つ一つの文章に名前がついており、入力された時間などの情報も表示される。オブジェクト分割表示を行う場合には、1 7 0 1 の B O X 文章一覧で所望の文章を選択して、オブジェクト表示ボタン 1 7 0 3 を押すことで表示が変わり、オブジェクト分割表示を行うことができる。これに関しては、後述する。また、1 7 0 1 の B O X 文章一覧で原稿を選択して、ページ表示ボタン 1 7 0 4 を押すことで表示を変えることができるが、これに関しても後述する。

#### 【 0 1 0 5 】

図 1 8 は、特に、保存された原稿のデータに基づいて原稿の画像を U I 上のページ表示画面 1 8 0 1 に表示したものを示している。ここでは、原稿のラスト画像を縮小した画像を表示することや、前述した S V G を使って表示をさせる事も可能である。つまり、前述してきたデータを基にページ全体を表示していれば良い。1 8 0 2 は、M F P 1 0 0 がもつ機能を選択するためのタブで、コピーや送信 ( F A X ) 、リモート操作、ブラウジング、B O X といった M F P 1 0 0 が持っている機能を選択できるものである。これ以外の機能に対しても同様の表示を行うことが出来ることは示すまでもない。

#### 【 0 1 0 6 】

同図の 1 8 0 3 により、原稿を読み取る場合の各種原稿モードを選択することが出来る。これは原稿タイプによって、画像処理を切り替えるために選択をするものであり、ここに示した以外のモードも同様に表示選択をする事が出来る。1 8 0 4 に示す読み込みスタートボタンにより、スキャナーが動作して、原稿の画像を読み込むことになる。この例では、読み込みスタートボタンを U I の画面内に設けているが、別途設けられたスタートボタンの押下によって原稿読み込みを開始してもよい。

#### 【 0 1 0 7 】

図 1 9 は、オブジェクト分割した結果が分かるように原稿内の各オブジェクトに枠を付けて表示させるようにしたものである。同図 1 9 0 1 のオブジェクト表示ボタンを押すことにより、ページ表示画面 1 9 0 2 に対してそれぞれオブジェクトの枠が表示される。このとき、枠に色付けをすることにより、オブジェクトの違いを分かるように表示することや、線の太さ、あるいは、点線、破線の違いなどにより、オブジェクトの違いを分かるように表示を行う。ここでオブジェクトの種類としては、前述したように文字、図面、線画、表、写真等である。

#### 【 0 1 0 8 】

同図の 1 9 0 3 は検索を行うための文字を入力する表示部分である。ここに検索文字列を入力して、検索を行うことで、検索対象のオブジェクトあるいはオブジェクトが含まれるページが検索される。検索方法に関しては、前述したオブジェクトに付与されたメタデータを利用し、周知の検索手法を用いることで、オブジェクトあるいはページの検索を行

10

20

30

40

50

うことができる。また、検索されたオブジェクトあるいはページは、UI上に表示される。

#### 【0109】

図20は、オブジェクト表示ボタン2002を押すことによりページ内のオブジェクトが表示されるものである。ここでのオブジェクトの表示は、ページという概念ではなく、2001に示されるように一つ一つのオブジェクトが部品として表示される。また、ページ表示ボタン2004を押すことで1ページの画像として見えるように表示することもできる。

#### 【0110】

同図の2003は検索を行うための検索文字列を入力する表示部分である。ここに検索文字列を入力して、検索を行うことで、検索対象のオブジェクトあるいはオブジェクトが含まれるページが検索される。検索方法に関しては、前述したオブジェクトに付与されたメタデータを利用し、周知の検索手法を用いることで、オブジェクトあるいはページの検索を行うことができる。また検索されたオブジェクトあるいはページは、UI上に表示される。

10

#### 【0111】

図21は、オブジェクトのメタデータをUI上に表示する例である。ある一つのオブジェクトを選択すると、2101に示すように、そのオブジェクトの画像2103と前述したメタデータ2102が表示される。ここで表示されるメタデータには、エリアの情報、幅、高さ、ユーザー情報、MF P 100の設置場所の情報、入力した時間などの情報が含まれている。

20

#### 【0112】

この例では、選択されたオブジェクトがpicture(写真)属性のオブジェクトであり、このオブジェクトの近くにあった文字オブジェクトの文字認識結果から形態素解析を用いて名詞のみを取り出して表示を行っている。それが図示したTEXTという文字列にあたる。また、メタデータに対しては、2104に示すように、ボタン操作により、編集や追加、削除を行うことを可能としている。

#### 【0113】

同図に示す2105は検索を行うための文字を入力する表示部分である。ここに検索文字列を入力して、検索を行うことで、検索対象のオブジェクトあるいはオブジェクトが含まれるページが検索される。検索方法に関しては、前述したオブジェクトに付与されたメタデータを利用し、周知の検索手法を用いることで、オブジェクトあるいはページの検索を行うことができる。また検索されたオブジェクトあるいはページは、UI上に表示される。

30

#### 【0114】

以上、述べてきたように、画像読み取り装置で読み取った画像データだけでなく、PDLデータをラスタライズした画像データも同様に、オブジェクト分割処理とベクトル化処理およびOCR処理を行っている。これにより、OCR処理結果をもとにしたメタデータを生成するので、保存する文書データのフォーマットが統一できる。また同時に、PDLデータ中のテキストオブジェクトから文字コードを抽出してメタデータとして使用しようとする場合に生じる、以下の問題を解決することができる。すなわち、

- ・見た目はテキストデータであるにも係わらず、内部で保持しているデータとしては、Pathデータやイメージデータである場合に文字コードが得られない、
- ・文字コードだけでは意味が不明なものが得られてしまう、
- ・もとのテキストの文字列が一文字一文字に分割されて描画コマンドになっている場合には、文字認識後の形態素解析で有意なテキストデータが得られない、

といった問題を解決することができる。

40

#### 【0115】

##### <第二の実施形態>

上述の第一の実施形態では、PDLデータがBOXに保存される際に、通常のレンダリ

50

ングが行われて、そのレンダリング結果をOCR処理する例を説明した。

#### 【0116】

この処理の際、文字データに対して、特別な文字修飾や特別なフォントが使用されていた場合には、OCR処理の文字認識精度が不十分となる場合が考えられる。そこで本実施形態においては、特にPDLデータのBOX保存に関して、さらにOCR処理の精度を高めるために、OCR処理専用のレンダリングを行う例を説明する。

#### 【0117】

本実施形態においては、MFP100はドライバUI600でPDLデータに対するBOX保存を指示された際に図22に示す各処理ブロックによる処理の流れに従って動作するようにする。

10

#### 【0118】

図22に示すように、PDL処理部304は、テキストオブジェクト変換処理部2201をさらに備えている。レンダラ305は、通常のレンダリングを行ってBitmapスプール201に画像を保存するパスとともに、OCR処理専用のレンダリングを行ってOCR専用Bitmapスプール2202に画像データを保存する構成をさらに備える。その他の処理ブロックは第一の実施形態において図4を用いて説明したとおりである。ただし、本実施形態においては、画像格納処理部200の構成、および、全体の処理フローが前述の第一の実施形態とは異なっている。

#### 【0119】

以下、図23と図24を用いて、本実施形態における画像格納処理部200の詳細な構成と、画像データ格納処理における処理フローについて説明する。

20

#### 【0120】

図23は、本実施形態における画像格納処理部200の構成を示すブロック図であり、図24は、PDLデータ保存処理の処理フローを示すフローチャートである。

#### 【0121】

画像格納処理部200において、図23に示すように、706のオブジェクト分割処理2部は、OCR専用Bitmapスプール2202からの出力を直接受けつける。ここでの処理は、オブジェクト分割処理部701と同様の処理を行えばよいが、特に、文字画像ブロックのみを抽出する処理を行う。オブジェクト分割処理2部706で判別された文字画像ブロックは、OCR処理部704において、第一の実施形態の場合と同様の処理を行い、文字コード情報を抽出する。

30

#### 【0122】

オブジェクト分割処理部701でも、第一の実施形態と同様にオブジェクト分割を行うが、本実施形態においては、文字ブロックの情報および画像データは、ベクトル化処理部703でのみ処理されるものとなる。各オブジェクトごとに、JPEG圧縮処理部702、ベクトル化処理部703、OCR処理部704による処理が行われた結果は統合されて、メタデータ付処理部705に入力されメタデータが付与されて、BOX保存される。

#### 【0123】

続いて、これら処理ブロックによって、BOX保存を行う際の処理フローの例を図24のフローチャートに示し、さらに詳細を説明する。

40

#### 【0124】

MFP100はクライアントPC102上で、ドライバUI600の指示に従って、PDLデータのBOX保存指示を受けて処理を開始する(S2400)。

#### 【0125】

ステップS2401でPDLデータをネットワークI/F114を介して受信して、PDLデータスプール303にデータを保存する。

#### 【0126】

続いて、保存したデータを取り出して、ステップS2402においてPDL処理部304およびレンダリング部305の処理ブロックによる処理を行う。

#### 【0127】

50

続いて、ステップ S 2 4 0 3 において、処理後の B i t m a p データを B i t m a p スプール 2 0 1 に保存する。この後、画像格納処理部 2 0 0 に処理を移す。すなわち、ステップ S 2 4 0 4 において、B i t m a p データを 7 0 1 のオブジェクト分割処理部により、オブジェクト分割する。

【 0 1 2 8 】

さらに、ステップ S 2 4 0 5 において、B i t m a p データのイメージ部分を J P E G 圧縮すると同時に、ステップ S 3 2 0 6 で、グラフィックス部分および文字部分に対するベクトル化処理部 7 0 3 による処理を実行させる。

【 0 1 2 9 】

他方において、P D L データに対してテキストオブジェクト変換処理部 2 2 0 1 による処理を実行させる。本実施形態において、テキストオブジェクトは図 2 5 に示すような表現がなされており、オブジェクトの座標・サイズ、文字コード列、フォント情報、および、文字修飾情報からなっている。

【 0 1 3 0 】

テキストオブジェクト変換処理部 2 2 0 1 では、フォント情報すなわちフォント種別とフォントサイズを O C R 処理が得意とするものに統一化して置き換える処理を行う。このことにより、O C R 処理が扱わなければならない多種のフォントを一つに統一し、従って、フォントが多種であることによって低下する可能性のある O C R 処理精度を高めることができる。また、テキストオブジェクト変換処理部 2 2 0 1 は、イタリック体やボールド体のような文字修飾情報を全て破棄するように動作する。このことによって、O C R 処理が多種の字体修飾を処理しなければならないことによる O C R 認識精度の低下を防ぐことが可能となる。

【 0 1 3 1 】

さらに付け加えれば、テキストオブジェクト変換処理部 2 2 0 1 で変換された P D L データは O C R 処理のためだけに使用され、画像記録装置 1 1 3 により記録される画像や U I 画面に表示される画像には影響を与えない。従って、これらの出力は、図 2 6 に示すように、変換前の全く見え方の異なる文字要素となる。2 6 0 1 のようにサイズが大きすぎて O C R による文字認識が成功しにくいもの、あるいは、パスデータとして処理してしまう可能性のあるものは、予め文字のサイズを小さくすることにより、O C R 処理の文字認識率を向上させることができる。

【 0 1 3 2 】

また、2 6 0 2 のように、フォントや文字修飾や回転がかかっている O C R 処理により文字認識が成功しにくいものや、あるいは、イメージデータとして処理してしまう可能性のあるものが存在する。このような文字に対しては、予め文字修飾情報を破棄したり、フォントの文字種およびサイズを統一化することで O C R 処理の文字認識率を向上させることができる。

【 0 1 3 3 】

また、2 6 0 3 のように、文字コードとフォントの組み合わせでレンダリングされて初めて意味のある視覚情報となるようなオブジェクトは、フォントを統一することで、文字コードが言語情報として意味のないものであることが分かることになる。このようにすれば、O C R 処理結果をもとにメタデータ付与する際の形態素解析において、不明語として処理されることにより、意味のないメタデータが付与されることを防ぐことができる。

【 0 1 3 4 】

以上のようにして変換された P D L データは、ステップ S 2 4 0 8 で P D L 処理およびレンダリングがなされる。

【 0 1 3 5 】

そして、生成された B i t m a p データを、ステップ S 2 4 0 9 において、O C R 専用 B i t m a p スプール 2 2 0 2 に保存する。

【 0 1 3 6 】

ステップ S 2 4 1 1 において、O C R 処理部 7 0 4 は O C R 専用 B i t m a p スプール

10

20

30

40

50



2202からデータを取り出してOCR処理を行い、文字コードを出力する。

【0137】

ステップS2412では文字コード情報とベクトル化されたデータの統合を行う。そして、ステップS3213においてメタデータ付与処理を行ってBOXへの保存を行う。

【0138】

以上の処理を行って、全体の処理を終える(S2414)。

【0139】

通常BitmapのレンダリングとOCR専用Bitmapのレンダリングの実行タイミングについては、本実施形態での処理フローではほぼ同時であるものとする。しかし、2つのBitmapスプールにそれぞれ画像データを保持することができるので、このタイ

10

【0140】

< 第三の実施形態 >

前述の第一の実施形態および第二の実施形態においては、文字コード情報としてOCR処理の結果から得られるものを使用してメタデータの付与を行う例を説明した。特に第二の実施形態においては文字認識精度を高めるべく、OCR専用のBitmapデータを生成する構成を説明した。本実施形態においては、OCR処理によって得られた文字コード情報と、PDLデータからそのまま抽出される文字コード情報を併用して文字コード情報の信頼性を高める例を図27から29を用いて説明する。

20

【0141】

図27は、本実施形態におけるPDLデータに対するBOX保存処理の際の各処理ブロックによる処理の流れを示す図である。図28は、同実施形態における画像格納処理部200の構成を示す図であり、図29は、文字コード情報統合処理部707が行う処理フローの例を示すフローチャートである。

【0142】

本実施形態においては、図27に示すように、レンダラ305に渡されるディスプレイリスト(Display List)情報から、文字コードを抽出する処理を行う文字コード情報抽出処理部2701を追加する。本実施形態の文字コード情報抽出処理部2701は、図25に示したテキストオブジェクト情報から座標・サイズ情報と文字コード情報を抽出し、画像格納処理部200に渡すように処理を行う。図27に示すその他の処理ブロックは、第二の実施形態で説明に用いた図22に示した対応する処理ブロックと同様のものとするが、第一の実施形態で説明に用いた図7に示した対応する処理ブロックと同様のものとしてもかまわない。

30

【0143】

画像格納処理部200は、図28に示すように、文字コード情報2を直接受け取り、文字コード情報2とOCR処理部704の処理によって得られた文字コード情報との統合を行う文字コード統合処理部707をさらに備えている。

【0144】

ここで、文字コード情報統合処理部707が行う処理の詳細を、図29を用いて説明する。

40

【0145】

文字コード情報統合処理が開始されると(S2900)、ステップS2901において、OCR処理部704の処理によって得られた文字コード情報と文字コード情報抽出処理部2701で抽出された文字コード情報2の対応関係の照合を行う。

【0146】

次に、ステップS2902において、位置・サイズが概略同じ文字コード情報を抽出する。

【0147】

50

次いで、ステップ S 2 9 0 3 でそれぞれの文字コード情報に含まれる文字コード列の形態素解析を行う。

【 0 1 4 8 】

そして、ステップ S 2 9 0 4 の形態素解析で得られた結果に不明語が少ないほうの文字コード情報を採用してこれを出し、全体の処理を終える ( S 2 9 0 4 )。

【 0 1 4 9 】

以上の処理により、得られる文字コード情報の信頼性を向上することが可能となる。

【 0 1 5 0 】

どちらの文字コードを出力するかを決定するルールとしては、以下のようなものを考えることもできる。すなわち、テキストオブジェクトが一文字一文字ばらばらに分解され P D L データとして送られてくる場合には、文字コード情報 2 は一文字のものばかりを含んでしまい、形態素解析の結果はほとんど意味のないものとなる。従って、文字コード列が長い方を採用するというルール付けをおこなってもよい。

【 0 1 5 1 】

< 第四の実施形態 >

前述の第三の実施形態で M F P 1 0 0 側に導入したテキストオブジェクト変換処理部 2 2 0 1 による処理は、プリンタドライバ側で行ってもよい。

【 0 1 5 2 】

図 3 0 は、クライアント P C 1 0 2 側の構成としてテキストオブジェクト変換処理 2 部 3 0 0 1 を追加した例を示している。テキストオブジェクト変換処理 2 部 3 0 0 1 の処理内容は、テキストオブジェクト変換処理部 2 2 0 1 と同様の処理を行う。

【 0 1 5 3 】

ドライバ U I 6 0 0 で P D L データの B O X 保存の指定がなされた際には、プリンタドライバ 3 0 2 からの P D L データ送信は、テキストオブジェクト変換を行わない通常の P D L データの送信となる。次に、O C R 専用のレンダリングを行う指定とともに、テキストオブジェクト変換処理がなされた P D L データを送信する。

【 0 1 5 4 】

M F P 1 0 0 側では、プリンタドライバ 3 0 2 からの一回目の P D L データ受信の際には、通常の B i t m a p スプール 2 0 1 に画像データが展開されるように制御する。そして、2 回目の P D L データ受信の際には、O C R 専用 B i t m a p スプール 2 2 0 2 に P D L データが展開されるように制御する。

【 0 1 5 5 】

通常の B i t m a p スプール 2 0 1 と O C R 専用 B i t m a p スプール 2 2 0 2 へ画像データの展開がなされた後の処理については、前述の第二あるいは第三の実施形態で説明した処理と同様に処理を行えばよい。

【 0 1 5 6 】

また、本実施形態においては、P D L 処理部 3 0 4 に接続されるテキストオブジェクト変換処理部 2 2 0 1 は必ずしも必要ではなくなる。

【 0 1 5 7 】

本実施形態の構成によれば、クライアント P C 1 0 2 側でアプリケーションとプリンタドライバの処理によって、もともとテキストデータであったものが、イメージデータになったり P a t h データになったりするケースを低減させることができる。このため、「見た目はテキストデータであるにも係わらず、内部で保持しているデータとしては、P a t h データやイメージデータである場合に文字コードが得られない」という問題の解決にさらに寄与できる。

【 0 1 5 8 】

< その他の実施の形態 >

前述の第一から第四の実施形態においては、P D L データが B O X に保存される際には、P D L データがレンダリングされた画像データをオブジェクト分割して、ベクトル化処理、J P E G 圧縮処理、O C R 処理がされるものとして説明してきた。しかし、図 3 1 に

10

20

30

40

50

示すように、PDL処理部304で生成されたDisplay List情報をそのまま画像格納処理部200に渡すようにしてもよい。

【0159】

このとき、MFP100は、テキストオブジェクト変換処理されていないPDLデータから生成されたDisplay Listをそのまま画像格納処理部200に渡すように制御する。そして、テキストオブジェクト変換処理を施したPDLデータはレンダラ305でOCR専用Bitmapデータとしてレンダリングされるように制御する。

【0160】

図32に示すように、画像格納処理部200は、入力されたDisplay Listが、3201においてSVGデータに直接変換されるように制御する。もともと、Display List上では、Imageオブジェクト、Graphicsオブジェクト、Textオブジェクトはオブジェクト分割されており、Graphicsオブジェクトはベクトル表現であるため、ある程度の直接変換が可能である。

10

【0161】

一方、これとは別に、OCR専用Bitmapデータから、オブジェクト分割処理部706によって得られた文字画像領域に対してOCR処理部704による処理を施す。そして文字コード情報を抽出し、第一から第四の実施形態で説明してきたようにメタデータ付与処理を行う。

【0162】

以上、説明してきたように、本実施形態においても、PDLデータ中の文字オブジェクトから文字コードを取り出して、それを検索のためのインデックスとする際の問題点を効果的に解決し、信頼性のある文書格納を行う画像処理装置を提供することが可能である。

20

【0163】

以上、本発明の諸実施形態について説明した。

【0164】

なお、本発明の目的は、上述した実施形態で示したフローチャートの手順を実現するプログラムコードを記憶した記憶媒体から、システムあるいは装置のコンピュータ（またはCPUやMPU）がそのプログラムコードを読み出し実行することによっても達成される。この場合、記憶媒体から読み出されたプログラムコード自体が、コンピュータに、上述した実施形態の機能を実現させることになる。そのため、このプログラムコード及びプログラムコードを記憶/記録したコンピュータ読み取り可能な記憶媒体も本発明の一つを構成することになる。

30

【0165】

プログラムコードを供給するための記憶媒体としては、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、CD-ROM、CD-R、磁気テープ、不揮発性のメモ리카ード、ROMなどを用いることができる。

【0166】

また、前述した実施形態の機能は、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実現される。また、このプログラムの実行とは、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う場合も含まれる。

40

【0167】

さらに、前述した実施形態の機能は、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よっても実現することもできる。この場合、まず、記憶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。その後、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う。こうした機能拡張ボードや機能拡張ユニットによる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】

50

## 【 0 1 6 8 】

【図 1】本発明に係る画像処理システムの概略構成を示すブロック図である。

【図 2】本発明の一実施形態である M F P の構成の例を示すブロック図である。

【図 3】画像読み取りの際の、同 M F P のコピー機能・スキャン送信機能・B O X 機能を実現する各処理ブロックによる処理の流れを示す図である。

【図 4】同 M F P の P D L データに対する P D L 処理機能・B O X 機能を実現する各処理ブロックによる処理の流れを示す図である。

【図 5】同実施形態におけるコピー・スキャン送信・B O X 保存等の機能を利用する際の U I の例を示す図である。

【図 6】同実施形態におけるドライバ U I の例を示す図である。

10

【図 7】画像格納処理部 2 0 0 の構成を示すブロック図である。

【図 8】メタデータ付与処理のフローチャートである。

【図 9】ベクトル化処理部 7 0 3 でベクトル化されたデータのフォーマットの一例である。

【図 1 0】オブジェクト分割処理による画像データのオブジェクト分割結果の一例を示す図である。

【図 1 1】オブジェクト分割したときの各属性のブロック情報および入力ファイル情報の例を示す図である。

【図 1 2】ベクトル化処理部 7 0 3 における処理の流れを示すフローチャートである。

【図 1 3】ベクトル化の処理における角抽出の処理を説明するための図である。

20

【図 1 4】ベクトル化の処理における輪郭線まどめの処理を説明するための図である。

【図 1 5】図 1 2 で生成されたベクトルデータのグループ化の処理を示すフローチャートである。

【図 1 6】図 1 5 の処理でグループ化されたベクトルデータに対する図形要素検出の処理を示すフローチャートである。

【図 1 7】同実施形態における B O X 保存時の U I 表示の一例を示した図である。

【図 1 8】同実施形態における B O X 保存時の U I 表示の例を示した図である。

【図 1 9】同実施形態における B O X 保存時の U I 表示の一例を示した図である。

【図 2 0】同実施形態における B O X 保存時の U I 表示の一例を示した図である。

【図 2 1】同実施形態における B O X 保存時の U I 表示のオブジェクトとメタ情報を表示される例を示した図である。

30

【図 2 2】第二の実施形態における P D L データに対する B O X 保存処理の際の各処理ブロックによる処理の流れを示す図である。

【図 2 3】同実施形態における画像格納処理部の構成を示すブロック図である。

【図 2 4】同実施形態における B O X 保存処理の処理フローを示すフローチャートである。

【図 2 5】同実施形態におけるテキストオブジェクトの表現例を示す図である。

【図 2 6】同実施形態におけるテキストオブジェクト変換処理を説明する図である。

## 【 0 1 6 9 】

【図 2 7】第三の実施形態における P D L データに対する B O X 保存処理の際の各処理ブロックによる処理の流れを示す図である。

40

【図 2 8】同実施形態における画像格納処理部 2 0 0 の構成を示す図である。

【図 2 9】同実施形態における文字コード情報統合処理部 7 0 7 が行う処理フローの例を示すフローチャートである。

【図 3 0】第四の実施形態におけるテキストオブジェクト変換処理のための構成例を示す図である。

【図 3 1】その他の実施形態における P D L データに対する B O X 保存処理の際の各処理ブロックによる処理の流れを示す図である。

【図 3 2】同実施形態における画像格納処理部 2 0 0 の構成を示す図である。

【符号の説明】

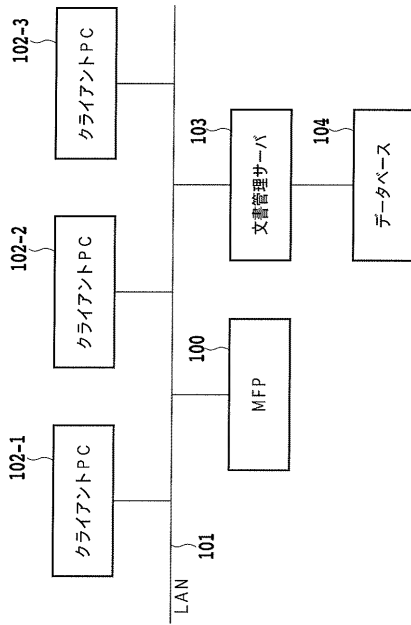
50

50

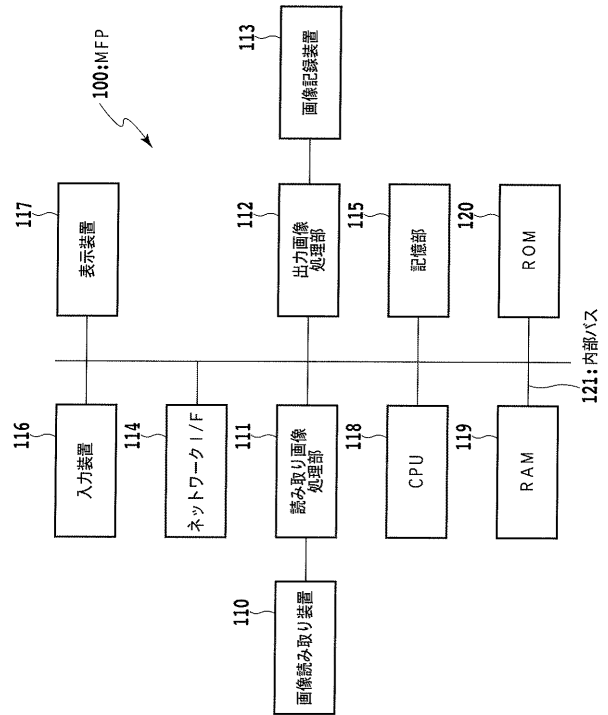
## 【 0 1 7 0 】

1 0 0	M F P	
1 0 1	L A N	
1 0 2	クライアント P C	
1 0 3	文書管理サーバ	
1 0 4	データベース	
1 1 0	画像読み取り装置	
1 1 1	読み取り画像処理部	
1 1 2	出力画像処理部	
1 1 3	画像記録装置	10
1 1 4	ネットワーク I / F	
1 1 5	記憶部	
1 1 6	入力装置	
1 1 7	表示装置	
1 1 8	C P U	
1 1 9	R A M	
1 2 0	R O M	
2 0 0	画像格納処理部	
2 0 1	B i t m a p スプール	
2 0 2	画像送信処理部	20
3 0 1	アプリ ( アプリケーション )	
3 0 2	プリンタドライバ	
3 0 3	P D L データスプール	
3 0 4	P D L 処理部	
3 0 5	レンダラ	
7 0 1	オブジェクト分割処理部	
7 0 2	J P E G 圧縮処理部	
7 0 3	ベクトル化処理部	
7 0 4	O C R 処理部	
7 0 5	メタデータ付与部	30
7 0 6	オブジェクト分割処理 2 部	
7 0 7	文字コード統合処理部	
2 2 0 1	テキストオブジェクト変換処理部	
2 2 0 2	O C R 専用 B i t m a p スプール	
2 7 0 1	文字コード情報抽出処理部	
3 0 0 0	テキストオブジェクト変換処理 2 部	
3 2 0 1	D i s p l a y L i s t   ベクトル化データ変換部	

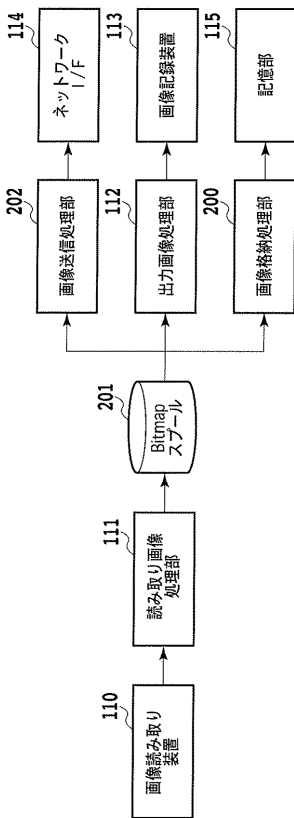
【図 1】



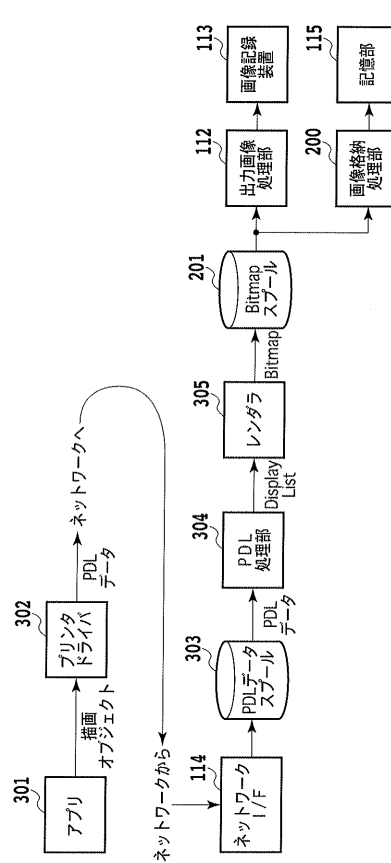
【図 2】



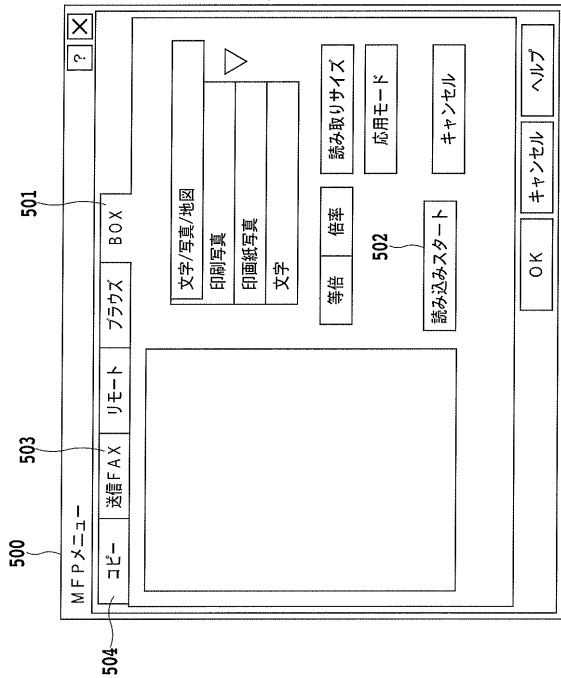
【図 3】



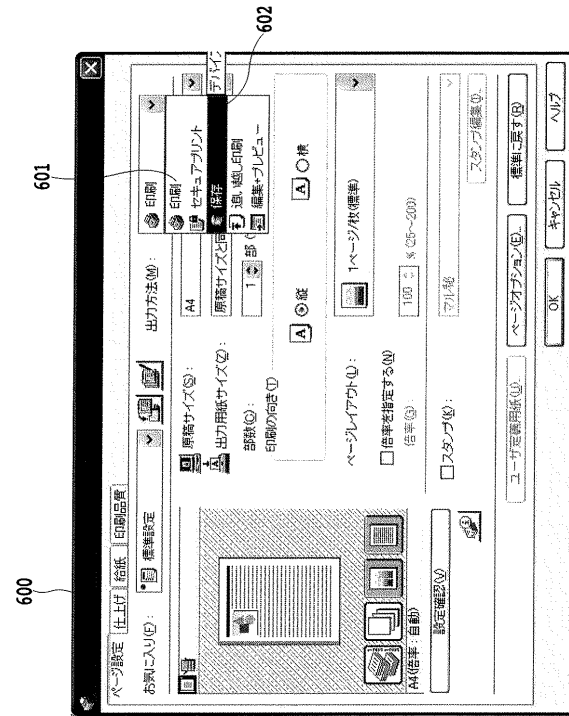
【図 4】



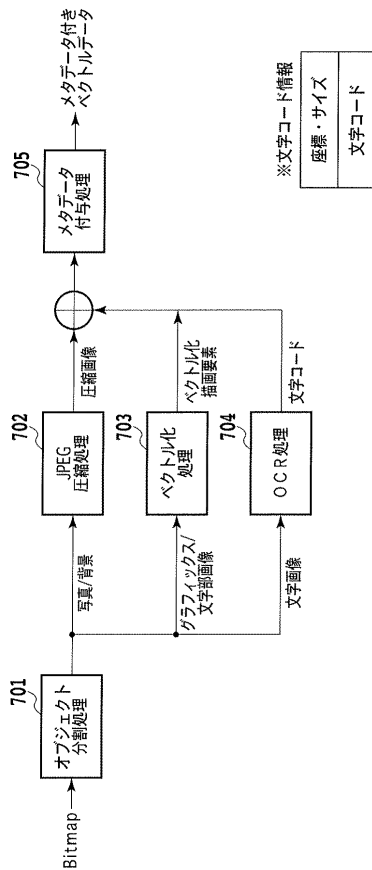
【図 5】



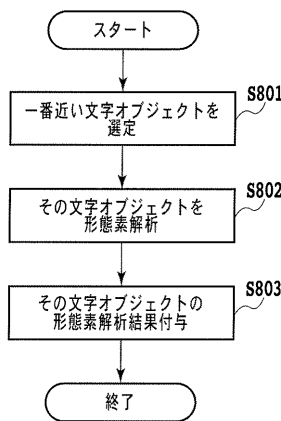
【図 6】



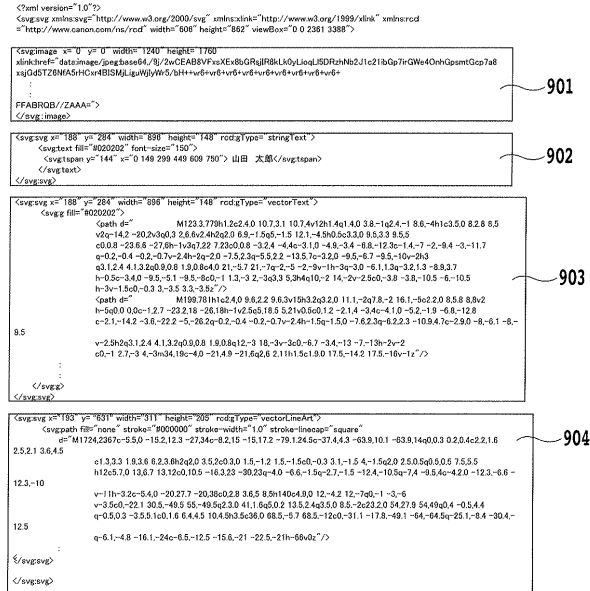
【図 7】



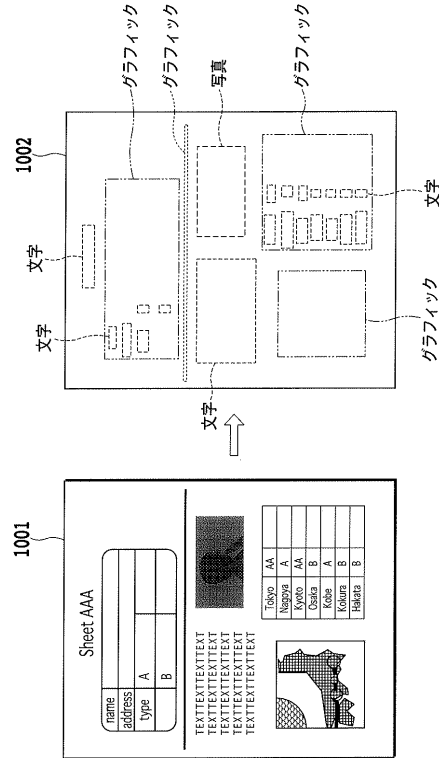
【図 8】



【図 9】



【図 10】



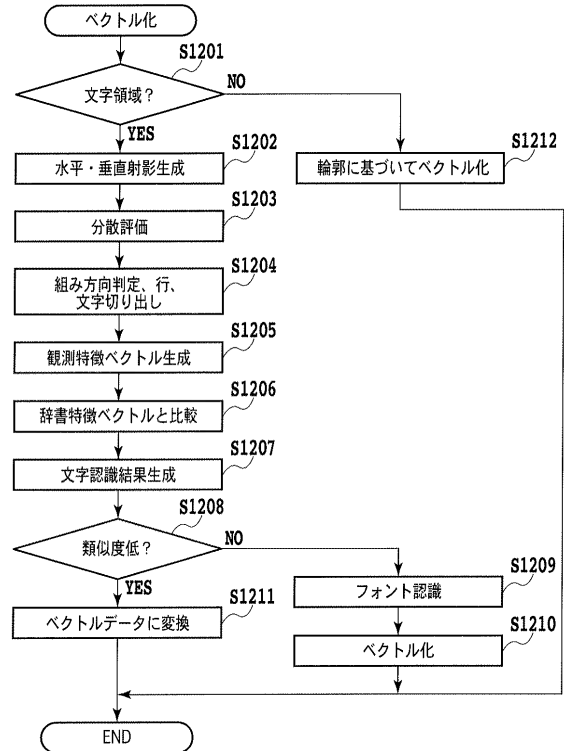
【図 11】

ブロック情報						OCR情報					
属性	座標 X	座標 Y	幅 W	高さ H		有	有	有	有	有	有
ブロック 1	X1	Y1	W1	H1		有	有	有	有	有	有
ブロック 2	X2	Y2	W2	H2		有	有	有	有	有	有
ブロック 3	X3	Y3	W3	H3		有	有	有	有	有	有
ブロック 4	X4	Y4	W4	H4		有	有	有	有	有	有
ブロック 5	X5	Y5	W5	H5		有	有	有	有	有	有
ブロック 6	X6	Y6	W6	H6		有	有	有	有	有	有

\*属性 1 : 文字, 2 : 写真, 3 : グラフィック  
入力ファイル情報

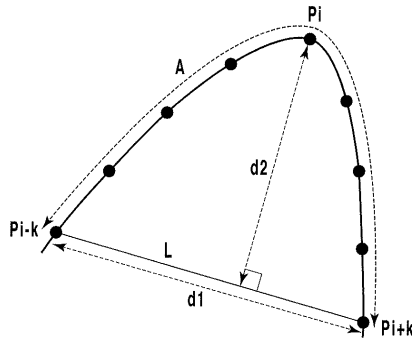
ブロック総数 N (=6)

【図 12】

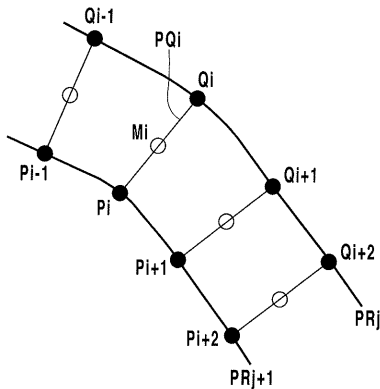




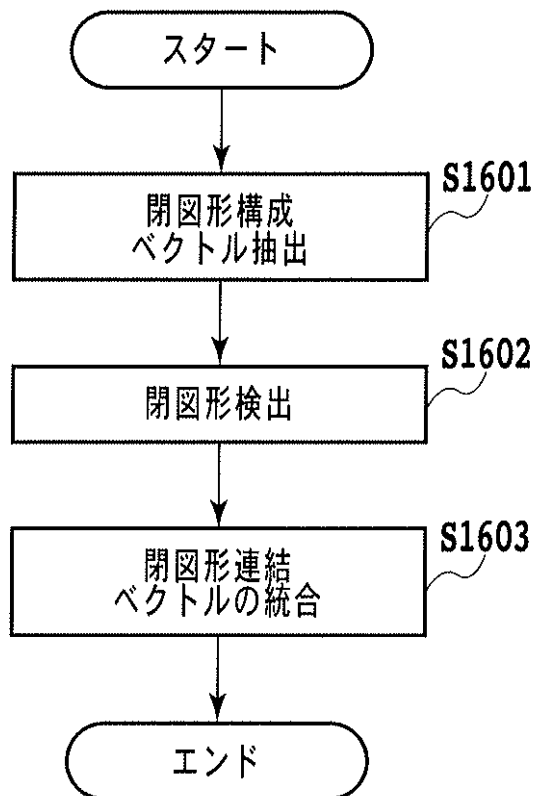
【図 13】



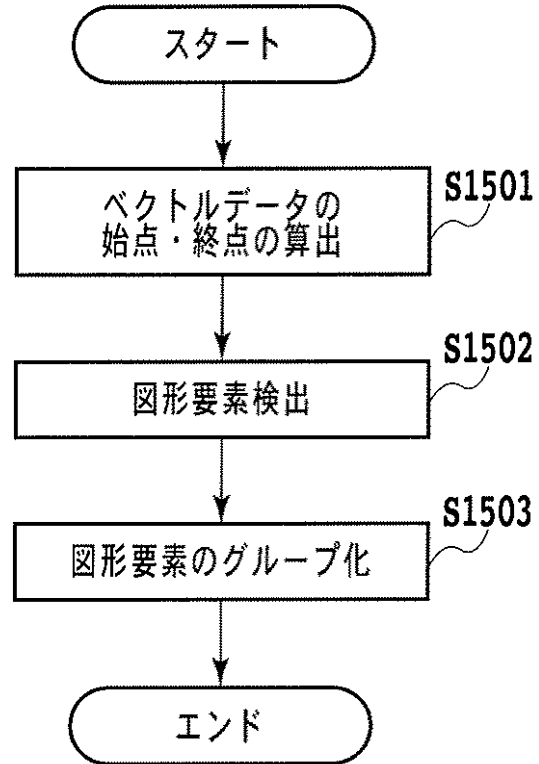
【図 14】



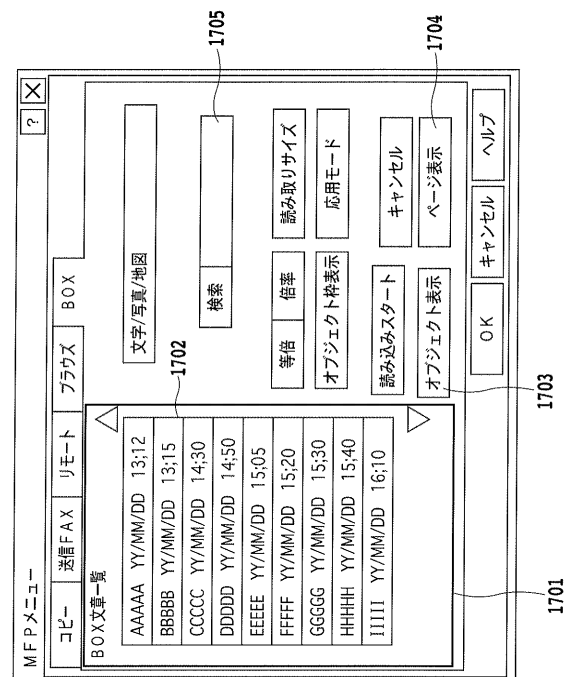
【図 16】



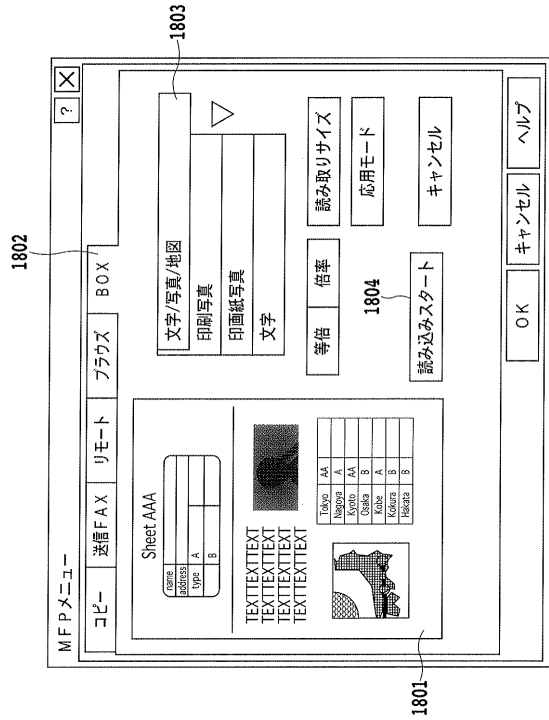
【図 15】



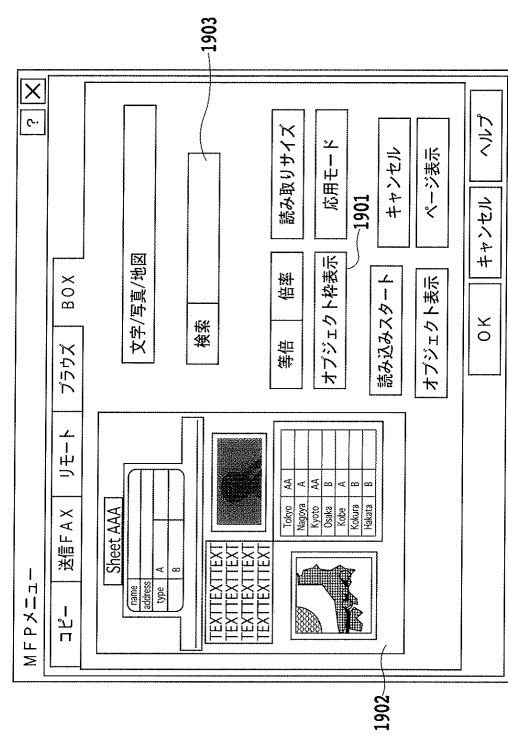
【図 17】



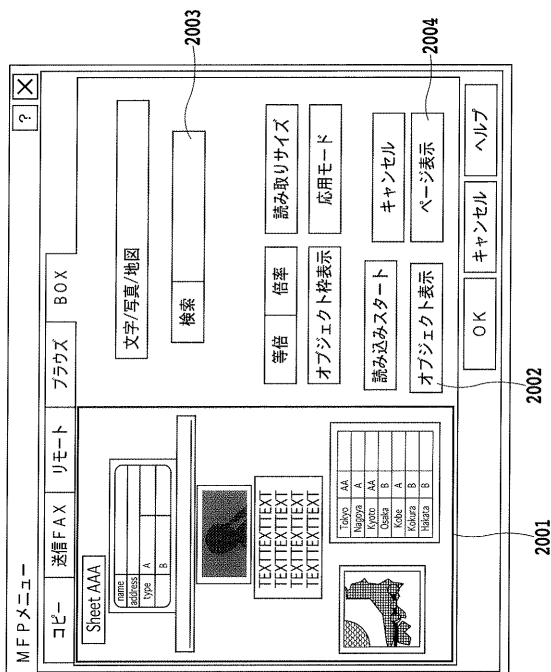
【図 18】



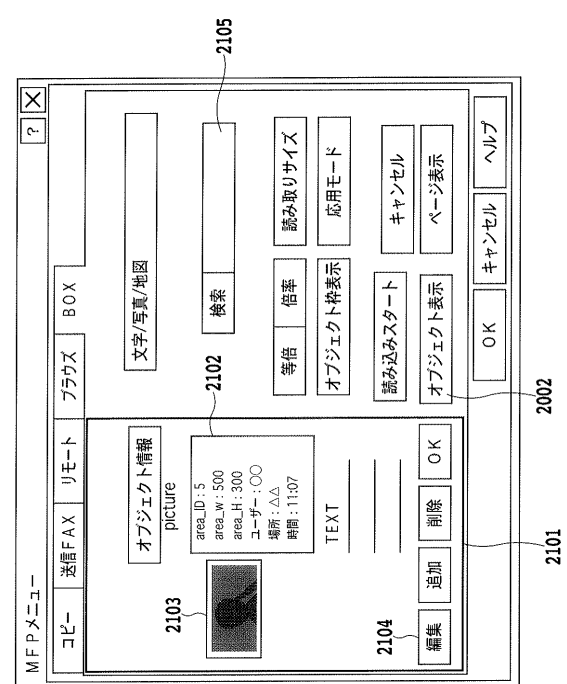
【図 19】



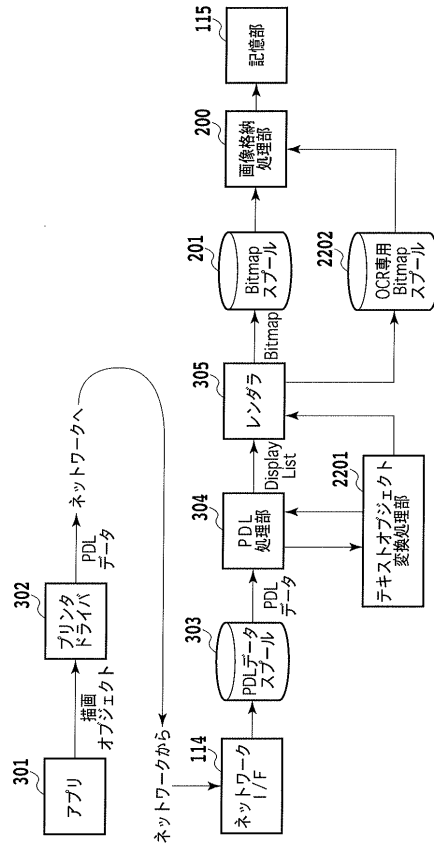
【図 20】



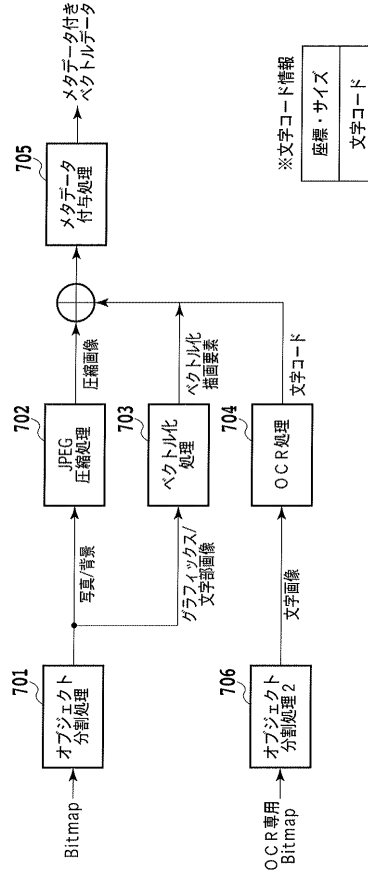
【図 21】



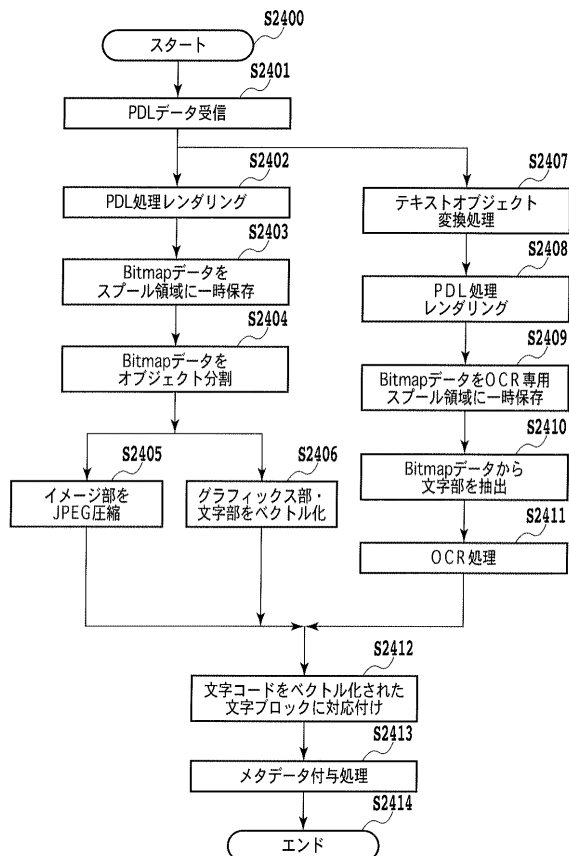
【図 22】



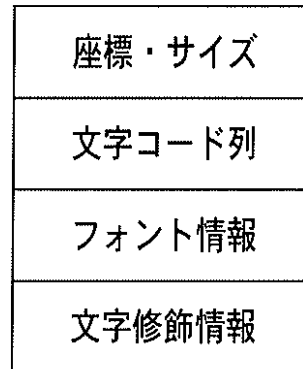
【図 23】



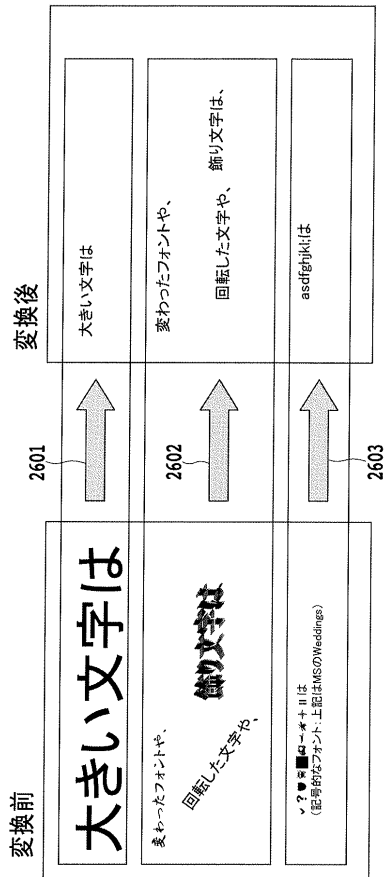
【図 24】



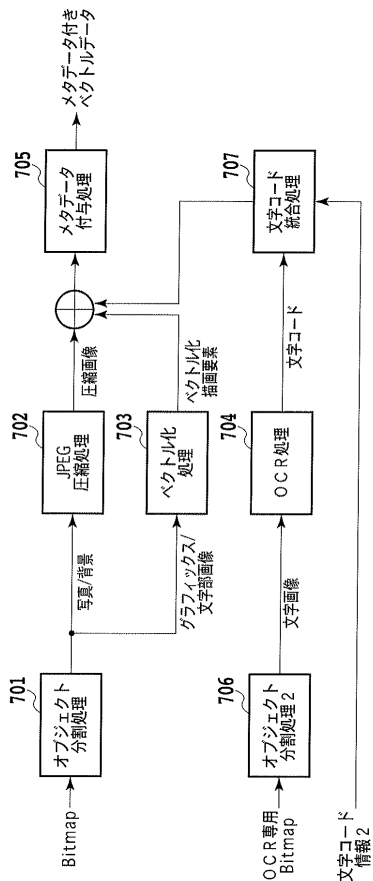
【図 25】



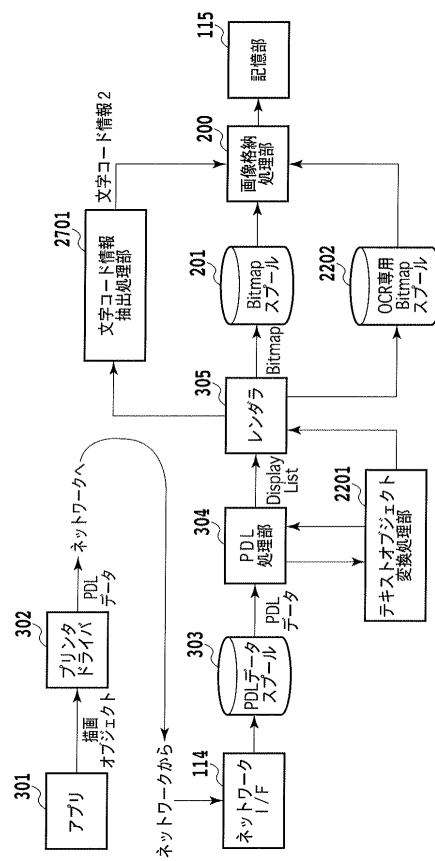
【図 26】



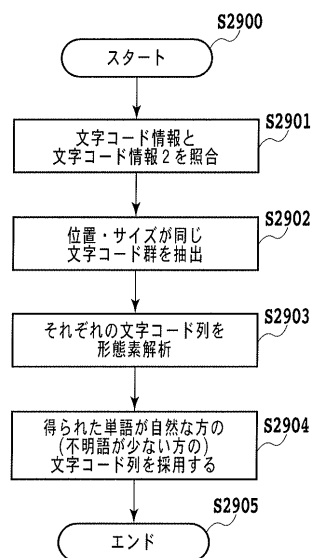
【図 28】



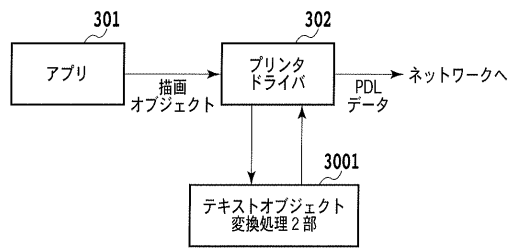
【図 27】



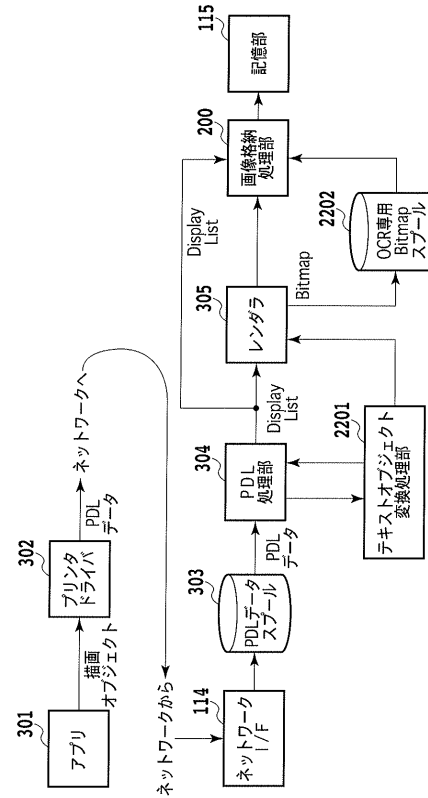
【図 29】



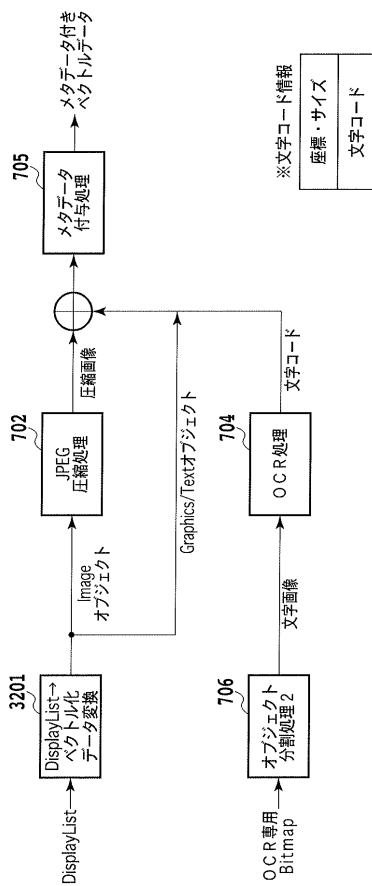
【 図 3 0 】



【 図 3 1 】



【 図 3 2 】



## フロントページの続き

- (72)発明者 竹林 学  
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 三沢 玲司  
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 飯沼 修  
東京都大田区下丸子3丁目30番2号 キヤノン株式会社内
- (72)発明者 伊藤 直樹  
東京都大田区下丸子3丁目30番2号 キヤノン株式会社内
- (72)発明者 檀淵 洋一  
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 荒川 純也  
東京都大田区下丸子3丁目30番2号 キヤノン株式会社内
- (72)発明者 佐野 真治  
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 真木 健彦

- (56)参考文献 特開2006-120125(JP,A)  
特開平10-228473(JP,A)  
特開2004-078531(JP,A)  
特開2004-272658(JP,A)  
平野 敬 Takashi HIRANO, ページ記述言語の解析に基づく多様な文書からの構造化内容情報の抽出 A Method of Logical Contents Extraction from Documents and Drawings Based on the Analysis of Page Description Language, 電子情報通信学会論文誌 (J91-D) 第5号 THE IEICE TRANSACTIONS ON INFORMATION AND SYSTEMS (JAPANESE EDITION), 日本, 社団法人電子情報通信学会 THE INSTITUTE OF ELECTRONICS, INFORMATION AND COMMUNICATION ENGINEERS, 2008年 5月 1日, 第J91-D巻 第5号, P.1406-1417

(58)調査した分野(Int.Cl., DB名)

G06T 1/00  
G06T 11/60  
G06T 11/80