(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2005-149097
(P2005-149097A)
(43)公開日　平成17年6月9日(2005.6.9)

| (51)Int.Cl.[7] | F I | | テーマコード（参考） |
|---|---|---|---|
| G06F 17/30 | G06F 17/30 | 170B | 5B050 |
| G06T 1/00 | G06F 17/30 | 240A | 5B075 |
| G06T 11/60 | G06T 1/00 | 200C | 5C073 |
| H04N 1/21 | G06T 11/60 | 100A | |
| | H04N 1/21 | | |

審査請求 未請求　請求項の数 14　ＯＬ　（全 28 頁）

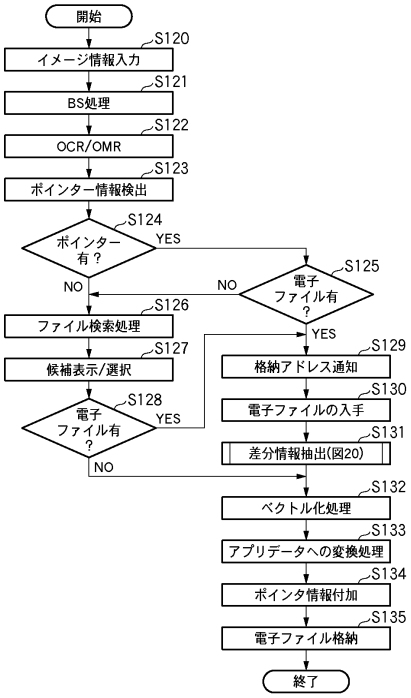
(21)出願番号　特願2003-385186 (P2003-385186)
(22)出願日　平成15年11月14日 (2003.11.14)

(71)出願人　000001007
キヤノン株式会社
東京都大田区下丸子３丁目３０番２号
(74)代理人　100076428
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　加藤　進一
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

最終頁に続く

(54)【発明の名称】画像処理システム及び画像処理方法

(57)【要約】
【課題】　ユーザが加筆した手書き文字等を含めた紙文書上の情報を消失することなく再利用・再編集可能な電子ファイルとして保存することができるとともに、それらを効率良く検索することができる画像処理システム及び画像処理方法を提供する。
【解決手段】　サーバが保持する電子ファイルの中から指定された原稿に対応する電子ファイルを取得して出力する。まず、原稿を表す画像データを入力し、入力された画像データに対応する電子ファイルをサーバが保持する電子ファイルの中から検索する。そして、検索された電子ファイルと入力された画像データとの差分情報を抽出し、電子ファイルと差分情報とを対応付けて所定の格納場所に格納する。
【選択図】　図３



【特許請求の範囲】
【請求項１】
サーバが保持する電子ファイルの中から指定された原稿に対応する電子ファイルを取得して出力する画像処理システムであって、
原稿を表す画像データを入力する入力手段と、
前記入力手段で入力された画像データに対応する電子ファイルを前記サーバが保持する電子ファイルの中から検索する検索手段と、
前記検索手段で検索された電子ファイルと前記入力手段で入力された画像データとの差分情報を抽出する抽出手段と、
前記電子ファイルと前記差分情報とを対応付けて保持する保持手段と
を備えることを特徴とする画像処理システム。
【請求項２】
前記入力手段が、所定の付加情報が形成された原稿を光学的に読み取り走査して得られた画像データを入力することを特徴とする請求項１に記載の画像処理システム。
【請求項３】
前記入力手段で入力された画像データから該画像データに対応する電子ファイルが格納されている格納場所を示す透かし、バーコード、文字、図形のいずれかで示されるポインタ情報を認識する認識手段をさらに備え、
前記検索手段は、前記認識手段で認識された前記ポインタ情報に基づいて前記画像データに対応する電子ファイルを検出する
ことを特徴とする請求項１又は２に記載の画像処理システム。
【請求項４】
前記差分情報を、その属性に応じて、文字コード化する文字処理手段、ベクトルデータに変換するベクトル化手段、所定の画像形式に変換する画像変換手段の少なくともいずれかを用いて変換する変換手段をさらに備えることを特徴とする請求項１から３までのいずれか１項に記載の画像処理システム。
【請求項５】
前記保持手段が、前記変換手段により変換された前記差分情報を前記電子ファイルと合成し、合成された新たな電子ファイルを合成前の電子ファイルに代えて保持することを特徴とする請求項４に記載の画像処理システム。
【請求項６】
前記保持手段が、前記変換手段により変換された前記差分情報を前記電子データと合成し、合成されたデータを新たな電子ファイルを保持することを特徴とする請求項４に記載の画像処理システム。
【請求項７】
前記保持手段が、前記変換手段により変換された前記差分情報を前記電子ファイルの既存の画像データの別レイヤーとして互いに関連付けて保持することを特徴とする請求項４に記載の画像処理システム。
【請求項８】
前記保持手段が、前記変換手段により変換された前記差分情報を前記電子ファイルとは別の差分情報ファイルとして保持することを特徴とする請求項４に記載の画像処理システム。
【請求項９】
前記保持手段が、前記電子ファイルに前記差分情報ファイルの保持先を示すポインタ情報を付加して該電子ファイルを保持することを特徴とする請求項８に記載の画像処理システム。
【請求項１０】
前記保持手段は、前記差分情報ファイルに前記電子ファイルの格納先を示すポインタ情報を付加して該差分情報を保持することを特徴とする請求項８に記載の画像処理システム。

【請求項１１】
アクセス制限のある画像データを保持するサーバから該画像データを取得するに先立って、認証情報の入力を要求するパスワード要求手段と、
入力された認証情報が正しいかどうかを判定する認証情報判定手段と、
正しい認証情報が入力された場合、前記サーバに対する前記画像データへのアクセスを許可する許可手段とをさらに備え、
前記許可手段によって前記画像データへのアクセスが許可されたことを条件として、該画像データを取得する
ことを特徴とする請求項１から１０までのいずれか１項に記載の画像処理システム。
【請求項１２】
前記抽出手段が、
前記検索手段で検索された前記電子ファイルと前記入力手段で入力された前記画像データとを比較して該画像データを補正する補正手段と、
補正後の画像データと前記電子ファイルとをブロックごとに同一性を比較する比較手段と、
前記比較手段により前記電子ファイルと同一のブロックであるとされたブロックを前記画像データから消去して前記差分情報を取得する消去手段と
を備えることを特徴とする請求項１から１１までのいずれか１項に記載の画像処理システム。
【請求項１３】
前記検索手段で検索された前記電子ファイルに前記ポインタ情報を付加する付加手段と、
前記ポインタ情報が付加された前記電子ファイルを所定媒体上に形成して出力する出力手段と
をさらに備えることを特徴とする請求項３に記載の画像処理システム。
【請求項１４】
サーバが保持する電子ファイルの中から指定された原稿に対応する電子ファイルを取得して出力する画像処理方法であって、
原稿を表す画像データを入力する入力工程と、
前記入力工程で入力された画像データに対応する電子ファイルを前記サーバが保持する電子ファイルの中から検索する検索工程と、
前記検索工程で検索された電子ファイルと前記入力工程で入力された画像データとの差分情報を抽出する抽出工程と、
前記電子ファイルと前記差分情報とを対応付けて所定の格納場所に格納する格納工程と
を有することを特徴とする画像処理方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、複写機等の画像処理装置で読み取った画像に基づいてデータベースに保存されている電子ファイルを検索する画像処理システム及び画像処理方法に関する。
【背景技術】
【０００２】
近年、環境問題がクローズアップされている中で、オフィス等でのペーパーレス化が急速に進んでいる。そこで、従来からバインダー等で蓄積・保存された紙文書をスキャナで読み取って、ポータブルドキュメントフォーマット（以下、「ＰＤＦ」と略す。）に変換し、画像記憶装置（データベース）に蓄積・保存する文書管理システムが構築されている。
【０００３】
一方、機能が拡張されたデジタル複合機（以下、「ＭＦＰ」と略す。）には、画像（イメージデータ）を紙文書として印刷・出力する際に、当該画像が記憶されている画像記憶

装置内のポインタ情報を紙文書の表紙或いは記載情報中に付加情報として記録しておくものがある。そして、当該画像が記録された紙文書を複写する場合には、表紙等に付加されているポインタ情報からオリジナルの画像が格納されている画像記憶装置内の格納場所を検出し、紙文書から読み取られた画像を再印刷等に用いるのではなく、画像記憶装置内に記憶されているオリジナルの画像を編集や再印刷に直接用いることができる。これにより、紙文書から読み取られた画像を何度も再利用することによって生じる画質の劣化等の問題を防ぐことができる（例えば、特許文献１参照。）。

【特許文献１】特開平１０－１４３４１４号公報

【発明の開示】

【発明が解決しようとする課題】

【０００４】

しかしながら、前者は紙文書をコンパクトな情報量のＰＤＦファイルとして保存することが可能であるが、ファイル全体が１つのイメージデータとして保存されるので、当該文書の一部のオブジェクトだけを部分的に検索して再利用・再編集することはできない。従って、当該データを再利用等する場合は、図や表等については、新たにアプリケーションソフトを用いて作成し直さなければならない。

【０００５】

また、後者は出力された紙文書に対応するオリジナルの電子ファイルに直接アクセスできるため容易に再利用・再編集することができる。しかし、紙文書上に後から手書き等によってユーザ等が追記した情報がオリジナルの電子ファイル中には存在しないため、紙文書から読み取ったイメージデータを対応するオリジナルの電子ファイルに置き換えてしまうことにより、追記された情報が消失してしまうという不具合が発生する。

【０００６】

本発明は、このような事情を考慮してなされたものであり、例えば、ユーザが加筆した手書き文字等を含めた紙文書上の情報を消失することなく再利用・再編集可能な電子ファイルとして保存することができるとともに、それらを効率良く検索することができる画像処理システム及び画像処理方法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】

【０００７】

上記課題を解決するために、本発明は、サーバが保持する電子ファイルの中から指定された原稿に対応する電子ファイルを取得して出力する画像処理システムであって、

原稿を表す画像データを入力する入力手段と、

前記入力手段で入力された画像データに対応する電子ファイルを前記サーバが保持する電子ファイルの中から検索する検索手段と、

前記検索手段で検索された電子ファイルと前記入力手段で入力された画像データとの差分情報を抽出する抽出手段と、

前記電子ファイルと前記差分情報とを対応付けて保持する保持手段と

を備えることを特徴とする。

【０００８】

また、本発明は、サーバが保持する電子ファイルの中から指定された原稿に対応する電子ファイルを取得して出力する画像処理方法であって、

原稿を表す画像データを入力する入力工程と、

前記入力工程で入力された画像データに対応する電子ファイルを前記サーバが保持する電子ファイルの中から検索する検索工程と、

前記検索工程で検索された電子ファイルと前記入力工程で入力された画像データとの差分情報を抽出する抽出工程と、

前記電子ファイルと前記差分情報とを対応付けて所定の格納場所に格納する格納工程と

を有することを特徴とする。

【発明の効果】

【０００９】

本発明によれば、ユーザが加筆した手書き文字等を含めた紙文書上の情報を消失することなく再利用・再編集可能な電子ファイルとして保存することができるとともに、それらを効率良く検索することができる。

【発明を実施するための最良の形態】

【００１０】

以下、図面を参照して、本発明の実施形態について説明する。

【実施例１】

【００１１】

図１は、本発明の第１の実施例に係る画像処理システムの構成を示すブロック図である。図１に示す画像処理システムは、オフィス１０とオフィス２０とをインターネット等のネットワーク１０４で接続された環境で実現される。

【００１２】

オフィス１０内に構築されたＬＡＮ１０７には、ＭＦＰ１００と、ＭＦＰ１００を制御するマネージメントＰＣ１０１と、クライアントＰＣ１０２と、文書管理サーバ１０６ａと、そのデータベース１０５ａ及びプロキシサーバ１０３ａが接続されている。また、オフィス２０内に構築されたＬＡＮ１０８には、文書管理サーバ１０６ｂと、そのデータベース１０５ｂ及びプロキシサーバ１０３ｂが接続されている。尚、クライアントＰＣ１０２は、外部記憶部、検索イメージ入力部及び検索結果出力部を備えている。また、ＬＡＮ１０７及びオフィス２０内のＬＡＮ１０８は、プロキシサーバ１０３ａ、１０３ｂを介してインターネット等のネットワーク１０４に接続されている。

【００１３】

ＭＦＰ１００は、本実施形態において紙文書を光学的に読み取って画像信号に変換する画像読み取り処理と、読み取った画像信号に対する画像処理の一部を担当し、画像信号はＬＡＮ１０９を用いてマネージメントＰＣ１０１に入力する。尚、マネージメントＰＣ１０１は、通常のＰＣでも実現可能であり、内部に画像記憶部、画像処理部、表示部及び入力部を備える。尚、マネージメントＰＣ１０１は、その一部又は全部をＭＦＰ１００と一体化して構成してもよい。

【００１４】

図２は、本発明の第１の実施例に係るＭＦＰ１００の構成を示すブロック図である。図２において、オートドキュメントフィーダ（以下、「ＡＤＦ」と略す。）を含む画像読み取り部１１０は、束状或いは１枚の原稿画像を内部に備える光源で照射し、原稿反射像をレンズで固体撮像素子上に結像し、固体撮像素子からラスタ状の画像読み取り信号を例えば６００ｄｐｉの密度のイメージ情報として得る。そして通常の複写機能を用いる場合は、この画像信号をデータ処理部１１５で記録信号へ画像処理し、複数毎複写の場合は記憶装置１１１に一旦１ページ分の記録データを保持した後、形成装置１１２に順次出力して紙上に画像を形成する。

【００１５】

一方、クライアントＰＣ１０２から出力されるプリントデータは、ＬＡＮ１０７からＭＦＰ１００に入力され、ネットワークＩＦ１１４を経てデータ処理装置１１５で記録可能なラスタデータに変換された後、形成装置１１２に出力して紙上に記録画像として形成される。

【００１６】

ＭＦＰ１００への操作者の指示は、ＭＦＰ１００に装備されているキー等の入力装置１１３、或いはマネージメントＰＣ１０１のキーボードやマウス等からなる入力装置から行われ、これら一連の動作はデータ処理装置１１５内の制御部で制御される。

【００１７】

一方、操作入力の状態表示及び処理中の画像データの表示は、ＭＦＰ１００の表示装置１１６又は、マネージメントＰＣ１０１、クライアントＰＣ１０２のモニタ等で行われる。尚、記憶装置１１１は、マネージメントＰＣ１０１からも制御され、ＭＦＰ１００とマネージメントＰＣ１０１とのデータの授受及び制御は、ネットワークＩＦ１１７及び直結

したＬＡＮ１０９を用いて行われる。
【００１８】
［処理概要］
次に、本発明の一実施形態に係る画像処理システムによる画像処理全体の概要について説明する。図３は、本発明の第１の実施例に係る画像処理システムによる画像処理の手順について説明するためのフローチャートである。
【００１９】
まず、ＭＦＰ１００の画像読み取り部１１０によって原稿画像の入力を行うために、当該画像読み取り部１１０を動作させて１枚の原稿をラスタ状に走査し、例えば６００ｄｐｉ、８ビットの画像信号を取得する（イメージ情報入力処理：ステップＳ１２０）。
【００２０】
入力された画像信号に対してデータ処理部１１５で前処理を施し、記憶装置１１１に１ページ分のイメージデータ（画像データ）として保存する。そして、マネージメントＰＣ１０１のＣＰＵは、記憶装置１１１に格納されたイメージデータから、文字／線画部分とハーフトーンの画像部分とに領域を分離し、文字部分はさらに段落で塊としてまとまっているブロック毎に、或いは、線で構成された表、図形に分離して各々セグメント化する。一方、ハーフトーンで表現される画像部分は、矩形に分離されたブロックの画像部分や背景部等のいわゆるブロック毎に独立したオブジェクトに分割する（ブロックセレクション処理：ステップＳ１２１）。
【００２１】
このとき、原稿画像中に付加情報として記録されている２次元バーコード、或いはＵＲＬに該当するオブジェクトを検出し、２次元バーコードの場合であれば当該マークを解読し、ＵＲＬの場合であればＯＣＲ（Optical Character Reader）等で文字認識する（ステップＳ１２２）。これによって、当該原稿画像のオリジナルの電子ファイルが格納されている記憶先のサーバ等のアドレスに関するポインタ情報を検出する（ステップＳ１２３）。尚、ポインタ情報を付加する手段は、上記の他に１次元バーコードや、文字と文字の間隔に情報を埋め込む方法及びハーフトーンの画像に埋め込む方法等の直接可視化されない電子透かしによる方法を用いてもよい。
【００２２】
次いで、ポインタ情報が検出できたかどうかを判定し（ステップＳ１２４）、ポインタ情報が検出された場合（Ｙｅｓ）は、検出されたポインタで示されたオリジナルの電子ファイルが格納されているサーバ等のアドレスに従って、図１で示されるネットワーク１０４又はＬＡＮ１０７等に接続されたサーバ等が保持する電子ファイルを検索する（ステップＳ１２５）。尚、本実施例では、電子ファイルは、図１で示したクライアントＰＣ１０２内のハードディスク内、或いはオフィス１０、２０のＬＡＮ１０７、１０８に接続された文書管理サーバ１０６ａ、ｂ内のデータベース１０５ａ、ｂ内、或いはＭＦＰ１００自体が有する記憶装置１１１のいずれかに格納されており、ステップＳ１２３で得られたサーバアドレス情報（ポインタ情報）に従ってこれらの記憶装置内を検索する。
【００２３】
ステップＳ１２５で特定の電子ファイルが見つからなかった場合や見つかったがＰＤＦ、ｔｉｆｆ等のいわゆるイメージファイルであった場合（Ｎｏ）、或いはステップＳ１２４でポインタ情報が検出できなかった場合（すなわち、ポインタ情報が示すサーバアドレス自体が存在しなかった場合）は、全サーバで管理されているデータ（すなわち、本実施例では、データベース１０５ａ、ｂ、記憶装置１１１に記憶されている全データ）を対象に入力されたイメージデータに基づいてステップＳ１２６以下のファイル検索処理を行う。一方、ステップＳ１２５でポインタ情報に基づくサーバアドレスからサーバが見つかった場合（Ｙｅｓ）は、ポインタが示すサーバを検索対象としてステップＳ１２９以下の処理を行う。
【００２４】
ステップＳ１２６は、いわゆる文書検索処理ルーチンである。まず、ステップＳ１２２

で各文字ブロックに対して行ったＯＣＲの結果から単語を抽出して、全文検索、或いは各オブジェクトの配列と各オブジェクトの属性からいわゆるレイアウト検索を行う。そして、検索の結果、類似度の高い電子ファイルが検索対象のサーバから見つかった場合は、その電子ファイルを候補としてそのサムネール等を表示し、複数の中から操作者の選択が必要なら操作者の入力操作によってファイルの特定を行う（ステップＳ１２７）。尚、候補が１ファイルの場合、ステップＳ１２７をバイパスして自動的にステップＳ１２８に分岐し、候補ファイルの格納アドレスをユーザに通知するようにしてもよい。一方、ステップＳ１２６の検索処理で電子ファイルが見つからなかった場合や見つかったがＰＤＦやｔｉｆｆ等のイメージファイルの場合（ステップＳ１２８でＮｏの場合）は、ステップＳ１３２に分岐する。

【００２５】

ここで、ステップＳ１２０で入力されたイメージ情報は、上記処理で検索、特定された電子ファイルをそのままプリント出力したものである場合だけでなく、プリント出力が資料として配付され、それを入手したユーザがその資料上にペン等で手書きによる書き込みをした場合や、元の特定された電子ファイルには存在しない新規の情報を付加した場合等も想定する。

【００２６】

そこで、ステップＳ１２５、Ｓ１２８で電子ファイルが特定された場合、入力されたイメージ情報と特定された電子ファイルとの差分情報を抽出して、オリジナルの原稿に対して後から追記された新規情報を抽出するようにする。すなわち、特定された電子ファイルの格納場所等のアドレスをクライアントに通知し（ステップＳ１２９）により、実際に電子ファイルを入手し（ステップＳ１３０）、入力イメージ情報と入手したオリジナルの電子ファイルとの差分情報を抽出する（ステップＳ１３０。

【００２７】

抽出された差分情報は、ベクトル化処理（ステップＳ１３２）によりイメージデータからベクトルデータに変換される。例えば、差分情報が、手書きで追記された文字情報等である場合は、ＯＣＲによるコード化又は輪郭追跡によってアウトライン化することによりベクトル情報に変換される。また、入力されたイメージデータが図形等の場合にも、ベクトルデータとしてベクトル情報に変換する。

【００２８】

ステップＳ１３２でベクトル化された差分情報は、ユーザからの指示に従って、新たな電子ファイルとして生成され保存される。以下に生成の例を記載する。

（例１）ベクトル化された差分情報とオリジナルの電子ファイルとを合成し、オリジナルの電子ファイルに上書き保存する。

（例２）ベクトル化された差分情報とオリジナルの電子ファイルとを合成し、新たな電子ファイルとして生成する。この場合、オリジナルの電子ファイルは別に保存されている。

（例３）ベクトル化された差分情報をオリジナルの電子ファイル内の別レイヤーとして保存する。

（例４）ベクトル化された差分情報をオリジナルの電子ファイルとは別の差分情報ファイルとして保存する。この場合、後述するステップＳ１３４と同様な処理、すなわち、オリジナルの電子ファイルに差分情報ファイルの位置を示すポインタ情報を埋め込むことで、オリジナルの電子ファイルと差分情報ファイルのリンクを行う。また、差分情報ファイルにオリジナルの電子ファイルの位置を示すポインタ情報を埋め込んでも良い。

【００２９】

上記（例１）によれば、オリジナルの電子ファイルに対して差分情報を付加した形で上書きするので、記憶されるデータの容量を削減し、また、新たな電子ファイルの増殖を防ぐことができる。しかしながら、電子ファイルのオリジナル性の確保ができなくなる。

【００３０】

上記（例２）は、オリジナルの電子ファイルのオリジナル性の確保をすることができる。しかしながら、差分情報のみ異なる同じような内容の電子ファイルが複数記憶されてい

くことになる。

【００３１】

上記（例３）は、オリジナルの電子ファイルのオリジナル性を確保しつつ、新たな電子ファイルの増殖を防ぐことができる。しかしながら、差分情報をレイヤー構造で保存するため、電子ファイルの構造が複雑になってしまう。

【００３２】

上記（例４）は、オリジナルの電子ファイルのオリジナル性を確保しつつ、差分情報ファイルを別に保存するので、差分情報ファイル専用の記憶手段等を設けておくことで、以後の検索の高速化を図ることができる。しかしながら、異なる差分情報が多い場合、差分情報ファイルが多く生成されてしまう。

【００３３】

上述したような差分情報とオリジナルの電子ファイルとの関係から、ユーザの使用に最適な形で差分情報と電子ファイルを生成し保存することで、ファイル容量を削減したり、検索を高速化したり、オリジナル性の確保を行うことができる。そこで、システムで予め差分情報の取り扱いについて固定しておく場合だけでなく、ユーザ側からどのような形で差分情報を取り扱うか、例えば、上記（例１）から（例４）のどの形で差分情報を格納するかを選択するようにしてもよい。

【００３４】

但し、ステップＳ１２８で電子ファイルの特定ができなかった場合（Ｎｏ）は、入力されたイメージデータ全体に対してベクトル化処理を行い、イメージデータをオリジナルの電子ファイルに近い電子ファイルに変換する。例えば、ステップＳ１２２でＯＣＲ処理された文字ブロックに対しては、さらに文字のサイズ、スタイル及び字体等を認識し、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。また、線で構成される表や図形ブロックに対しては、アウトライン化する。さらに、画像ブロックに対しては、そのままイメージデータとして個別のＪＰＥＧファイル等として処理する。

【００３５】

これらのベクトル化処理（ステップＳ１３２）は各オブジェクト毎に行い、更に各オブジェクトのレイアウト情報を保存して、一般のアプリケーションで編集可能なアプリデータに変換し（ステップＳ１３３）、新たな電子ファイルとして記憶装置１１１等に格納する（ステップＳ１３５）。また、ＪＰＥＧファイルではなく、汎用のファイル形式として、例えばＲＴＦ（Rich Text Format）形式に変換するようにしてもよい。

【００３６】

尚、本実施例では、ステップＳ１３３でのアプリデータへの変換処理後に、電子ファイルの格納場所のサーバ等のアドレスを「ポインタ情報（アドレス情報）」として、保存対象の電子ファイルに付加するようにする（ステップＳ１３４）。このようにすることにより、保存された電子ファイルが格納されたサーバ等（例えば、記憶装置１１１等）が次回以降の検索対象となる。また、電子ファイルの印刷時にポインタ情報を２次元バーコード等を用いて付加して印字するようにすれば、次回以降にポインタ情報が付加された原稿を読み取って、当該ポインタ情報を解析することにより、オリジナルの電子ファイルを保存するサーバ等に容易にアクセスすることができるようになる。

【００３７】

以上の手順によってオリジナルの電子ファイル、又はそれに非常に近いベクトル情報を有する電子ファイルを容易に検索して取得することができる。すなわち、本実施形態では、電子ファイルのデータはベクトル化されたデータであって、編集可能な形式で全て保存されており、従来のように１枚のイメージデータとして取得するだけではなく、取得後にそのデータを直接加工、再利用したり、あるいは蓄積、伝送、再印刷を行うことが可能になる。

【００３８】

これらの処理は、単純にイメージデータを直接扱う場合に比べて単に再利用・再編集が可能となるだけではなく、情報量を削減することができ、蓄積効率が高まり、伝送時間が

短縮され、記録表示する際には高品位なデータとして非常に優位となる。

【００３９】

以下、各処理ブロックに対して詳細に説明する。

【００４０】

［ブロックセレクション（ＢＳ）処理（ステップＳ１２１）］

図４は、ブロックセレクション処理によって読み取った１枚のイメージデータを属性を判定し複数のブロックに分割する様子を示す図である。すなわち、ブロックセレクション処理とは、符号４１に示すステップＳ１２０で読み取った一頁のイメージデータを、符号４２に示すようにオブジェクト毎の塊として認識し、それぞれのブロックを文字（Text）、写真（Photo）、線（Line）、表（Table）等の属性に判定し、異なる属性を持つ領域（ブロック）に分割する処理である。

【００４１】

ブロックセレクション処理の一実施例を以下に説明する。

【００４２】

まず、入力画像を白黒に２値化して、輪郭線追跡を行って黒画素輪郭で囲まれる画素の塊を抽出する。そして、面積の大きい黒画素の塊については、内部にある白画素に対して同様に輪郭線追跡を行って白画素の塊を抽出する。さらに、一定面積以上の白画素の塊の内部からも再帰的に黒画素の塊を抽出する。

【００４３】

このようにして得られた黒画素（白画素）の塊を、大きさ及び形状等で分類し、異なる属性を持つ領域へ分類する。例えば、縦横比が１に近く、大きさが一定の範囲のものを文字相当の画素塊とし、さらに近接する文字が整列良くグループ化可能な部分を文字領域とする。また、扁平な画素塊を線領域、一定の大きさ以上でかつ四角系の白画素塊を整列よく内包する黒画素塊の占める範囲を表領域、不定形の画素塊が散在している領域を写真領域、それ以外の任意形状の画素塊を図画領域等とする。

【００４４】

図５は、ブロックセレクション処理で得られた各ブロックに対するブロック情報の一例について示す図である。図５に示されるブロック毎の情報は、後述するベクトル化、検索処理のための情報として用いられる。

【００４５】

［イメージデータからポインタ情報の検出］

まず、ステップＳ１２２で示す電子ファイルの格納位置を、読み取られたイメージデータから抽出するためのＯＣＲ／ＯＭＲ処理について説明する。図６は、原稿画像中に付加された２次元バーコード（ＱＲコードシンボル）を復号してデータ文字列を出力する手順を説明するためのフローチャートである。図７は、２次元バーコードが付加された原稿３１０の一例を示す図である。

【００４６】

まず、データ処理装置１１５内のページメモリに格納された原稿３１０を読み取って得られたイメージデータを内部のＣＰＵで走査して、前述したブロックセレクション処理の結果から所定の２次元バーコードシンボル３１１の位置を検出する。ＱＲコードの位置検出パターンは、シンボルの４隅のうちの３隅に配置される同一の位置検出要素パターン３１１ａ～３１１ｃから構成される（ステップＳ３００）。

【００４７】

次に、位置検出パターンに隣接する形式情報を復元し、シンボルに適用されている誤り訂正レベル及びマスクパターンを得る（ステップＳ３０１）。さらに、シンボルの型番を決定した後（ステップＳ３０２）、形式情報で得られたマスクパターンを使って符号化領域ビットパターンをＸＯＲ演算することによってマスク処理を解除する（ステップＳ３０３）。

【００４８】

そして、モデルに対応する配置規則に従い、シンボルキャラクタを読み取り、メッセー

ジのデータ及び誤り訂正コード語を復元する（ステップＳ３０４）。次いで、復元されたコード上に誤りがあるかどうかの検出を行う（ステップＳ３０５）。その結果、誤りが検出された場合（Ｙｅｓ）は当該誤りを訂正する（ステップＳ３０６）。そして、誤り訂正されたデータより、モード指示子及び文字数指示子に基づいて、データコード語をセグメントに分割する（ステップＳ３０７）。最後に、仕様モードに基づいてデータ文字を復号し、結果を出力する（ステップＳ３０８）。尚、ステップＳ３０５で誤りが検出されなかった場合（Ｎｏ）は、上記ステップＳ３０７に進む。

【００４９】

ここで、２次元バーコード内に組み込まれたデータは、対応する電子ファイルが格納されているサーバアドレス情報（ポインタ情報）を表しており、例えばファイルサーバ名及びサーバアドレスを示すＩＰアドレス、対応するＵＲＬ等からなるパス情報で構成される。

【００５０】

本実施例では上述したように、ポインタ情報が２次元バーコードを用いて付与された原稿３１０を例に挙げて説明したが、直接文字列でポインタ情報が記録される場合には、所定のルールに従った文字列のブロックを前述したブロックセレクション処理で検出し、当該ポインタ情報を示す文字列の各文字を文字認識することで、直接、オリジナルの電子ファイルが保存されているサーバのサーバアドレス情報を得ることが可能である。

【００５１】

また、図７の文書３１０の文字ブロック３１２や文字ブロック３１３の文字列に対して、隣接する文字と文字の間隔等に視認し難い程度の変調を加え、当該文字間隔を用いた透かし情報を埋め込むことでもポインタ情報を付与することができる。このような透かし情報は、後述する文字認識処理を行う際に各文字の間隔を検出することによって、ポインタ情報を得ることができる。また、自然画ブロック３１４の中に電子透かしとしてポインタ情報を付加することも可能である。

【００５２】

［ポインタ情報による電子ファイル検索処理］

次に、図３を用いて説明したステップＳ１２５、Ｓ１２８の処理で行われるポインタ情報からの電子ファイル検索処理について詳細に説明する。図８は、検出されたポインタ情報から電子ファイルを検索する処理手順について説明するためのフローチャートである。

【００５３】

まず、ポインタ情報に含まれるサーバアドレスに基づいて、当該電子ファイルが格納されているファイルサーバを特定する（ステップＳ４００）。ここでファイルサーバとは、クライアントＰＣ１０２や、データベース１０５ａ、ｂを内蔵する文書管理サーバ１０６ａ、ｂや、記憶装置１１１を内蔵するＭＦＰ１００自身を指す。また、アドレスとは、ＵＲＬやサーバ名からなるパス情報である。

【００５４】

そして、ファイルサーバを特定した後、図３を用いて説明したステップＳ１２７におけるポインタ情報が示すサーバ（ファイルサーバ）に対してアドレスを転送する（ステップＳ４０１）。ファイルサーバは、ステップＳ１２６のファイル検索処理に従って、該当する電子ファイルを検索する（ステップＳ４０２）。そして、電子ファイルが存在するか否かを判定する（ステップＳ４０３）。

【００５５】

この結果、電子ファイルが存在しない場合（Ｎｏ）は、ＭＦＰ１００に対してその旨を通知して終了する。一方、電子ファイルが存在する場合（Ｙｅｓ）は、図３を用いて前述したように、差分抽出処理を行うために候補表示を行い、その後、該当する電子ファイルのアドレスを通知すると共に、当該電子ファイルをユーザ（すなわち、ＭＦＰ１００）に対して転送する（ステップＳ４０８）。

【００５６】

［ファイルアクセス権を含むポインタ情報によるサーバ検索処理］

次に、ファイルアクセス権について考慮した実施例について説明する。通常扱われる文書ファイルの中には、第三者による再利用を制限することが望ましい文書がある。前述した図８を用いた検索処理では、ファイルサーバに蓄積された電子ファイルは全て自由にアクセスすることが可能であり、ファイル全体、或いはその一部のオブジェクトは全て再利用が可能なことを前提に説明した。以下では、ポインタ情報から電子ファイルを検索した際に、検索の結果から特定された電子ファイルにアクセス権の制限が有る場合について説明する。

【００５７】

図９は、ファイルアクセス権を含むポインタ情報から電子ファイルが格納されているサーバを検索する処理手順について説明するためのフローチャートである。図９において、ステップＳ４００～Ｓ４０３までは、図８におけるステップと同様であるため、説明は省略する。ステップＳ４０３で電子ファイルが特定された場合（Ｙｅｓ）、ファイルサーバはそのファイルのアクセス権情報を調べる（ステップＳ４０４）。その結果、アクセス制限がある場合（Ｙｅｓ）は、ＭＦＰ１００に対してパスワードの送信を要求する（ステップＳ４０５）。

【００５８】

そして、ＭＦＰ１００は、操作者に対してパスワード等の認証情報の入力を促し、入力されたパスワード等をファイルサーバに送信する（ステップＳ４０６）。ファイルサーバは送信されたパスワード等を照合し（ステップＳ４０７）、一致した場合（Ｙｅｓ）は、図３を用いて説明したように、電子ファイルのアドレスを通知すると共に、ユーザの希望する処理がイメージデータの取得であれば、ＭＦＰ１００に対して電子ファイルを転送する（ステップＳ４０８）。

【００５９】

尚、アクセス権の制御を行うための認証の方法は、ステップＳ４０５、Ｓ４０６に示したパスワードによる方法に限定されず、例えば、指紋認証等の一般に広く用いられている生体認証、カードによる認証等のあらゆる認証手段を用いることができる。

【００６０】

一方、ステップＳ４０３でファイルサーバ内からファイルを特定することができなかった場合は、図３のステップＳ１３３で説明したベクトル化処理に対しても、制限を加えることができる。すなわち、紙文書を走査して得られたイメージデータからオリジナルの電子ファイルに対してのアクセス権の制限の存在を検出した場合には、認証確認が取れた場合のみベクトル化処理を行うことで、機密性の高い文書の使用に制限をかけることができる。

【００６１】

［ファイル検索処理］

次に、図３のステップＳ１２６で示すファイル検索処理の詳細について図５及び図１０を使用して説明する。ここでは、ステップＳ１２２のＯＣＲ／ＯＭＲ処理の結果、抽出された各ブロック及び入力ファイルが、図５に示す情報（ブロック情報、入力ファイル情報）を備えるものとする。本実施形態では、情報内容として、図５に示すように属性、座標位置、幅及び高さのサイズ、ＯＣＲ情報の有無を用いる。

【００６２】

属性は、さらに、文字、線、写真、絵、表等に分類される。尚、図５では説明を簡単にするため、ブロックは座標Ｘの小さい順（例えば、Ｘ１＜Ｘ２＜Ｘ３＜Ｘ４＜Ｘ５＜Ｘ６）に、ブロック１、ブロック２、ブロック３、ブロック４、ブロック５、ブロック６としている。また、ブロック総数は、入力ファイル中の全ブロック数であり、図５に示す例におけるブロック総数は６である。以下、これらの情報を使用して、データベース内から入力イメージファイルに類似した電子ファイルのレイアウト検索を行う手順について説明する。図１０は、データベース内から入力イメージファイルに類似した電子ファイルのレイアウト検索を行う手順について説明するためのフローチャートである。尚、データベースファイルも、図５に示す情報と同様の情報を備えることを前提とする。図１０のフローチ

ャートの流れは、入力された原稿から読み取られた電子ファイルとデータベース中の電子ファイルを順次比較するものである。

【００６３】

まず、後述する類似率等の初期化を行って初期値を設定する（ステップＳ５１０）。次に、ブロック総数の比較を行い（ステップＳ５１１）、真の場合（Ｙｅｓ）は、さらにファイル内のブロックの情報を順次比較する（ステップＳ５１２）。すなわち、データベースのファイルのブロック数ｎが入力ファイルのブロック数Ｎの誤差ΔＮ範囲内かどうかを調べ、誤差範囲内であれば真（Ｙｅｓ）、範囲外であれば偽（Ｎｏ）とする。また、ステップＳ５１２では、入力ファイルとデータベースファイルのブロック属性を比較して一致すればステップＳ５１３以降の比較処理へ進み、不一致であればステップＳ５２１に進む。

【００６４】

ブロックの情報比較では、ステップＳ５１３、Ｓ５１５、Ｓ５１８において、それぞれ属性類似率、サイズ類似率、ＯＣＲ類似率をそれぞれ算出し、ステップＳ５２２においてそれらに基づいて総合類似率を算出する。各類似率の算出方法については、公知の技術を用いることができるので説明を省略する。

【００６５】

ステップＳ５２３においては、総合類似率が予め設定された閾値Ｔｈより高いかどうかを判定し、高い場合（Ｙｅｓ）は、その電子ファイルを類似候補として挙げて保存する（ステップＳ５２４）。尚、図中のＮ、Ｗ、Ｈは、入力ファイルのブロック総数、各ブロック幅、各ブロック高さ、ΔＮ、ΔＷ、ΔＨは、入力ファイルのブロック情報を基準として誤差を考慮したものである。また、ｎ、ｗ、ｈは、データベースファイルのブロック総数、各ブロック幅、各ブロック高さとする。尚、ステップＳ５１４におけるサイズ比較時に、位置情報（Ｘ，Ｙ）の比較等を行ってもよい。

【００６６】

以上、検索の結果、総合類似率が閾値Ｔｈより高いもので候補として保存されたデータベースファイルをサムネール等で表示する（ステップＳ１２９）。これにより、複数の中から操作者の選択が必要な場合は、操作者の入力操作よってファイルの特定を行う。

【００６７】

［差分抽出処理］

次に、図３のステップＳ１３１で示す差分情報抽出処理の詳細について説明する。図２０は、図３に示すフローチャートにおける差分抽出処理（ステップＳ１３１）の手順を説明するための概略を示すフローチャートである。

【００６８】

図２０に示すフローチャートにおいて、「入力画像」とは図３のステップ１２０でイメージ情報として入力された原稿から読み取り走査されたイメージデータを示し、「電子ファイル」とは図３のステップＳ１２５、Ｓ１２８で得られたオリジナルの電子ファイルを示す。ここで入力画像は、図３のステップＳ１２１、Ｓ１２２、Ｓ１２３においてブロックセレクション処理、ＯＣＲ／ＯＭＲ処理、ポインタ情報の検出が終わっているので、図４で示したブロックセレクション後のブロックレイアウト情報、また、図５で示したブロック情報、図７で示したポインタ情報がすでに得られている。この様子を図２１に示す。

【００６９】

図２１は、図２０に示す差分抽出処理内容を説明するための電子ファイルに関する入力画像のイメージデータ例及びブロックセレクション例を示す図である。図２１において、２１１０はオリジナルの電子ファイルを示し、２１１１は電子ファイル２１１０に対してブロックセレクション処理を行った後のブロックレイアウト情報例を示す。尚、ブロックレイアウト情報としては、図５で示したブロック情報を別途含まれる。また、２１１４は図７で示したポインタ情報を示す２次元バーコードであり、２１１５はブロックセレクション処理後の２次元バーコードのブロックである。

【００７０】

また、図２１において、２１１２は入力画像の内容を示し、２１１３は入力画像２１１２に対してブロックセレクション処理を行った後のブロックレイアウト情報である。尚、ブロックレイアウト情報としては、図５で示したブロック情報を別途含まれる。また、２１１６は図７で示したポインタ情報を示す２次元バーコードであり、２１１７はブロックセレクション処理後の２次元バーコードのブロックである。ここで、電子ファイル２１１０とは異なり、入力画像２１１２には、手書き文字２１１８が記入されており、それをブロックセレクション処理することで、手書き文字ブロック２１１９がブロックレイアウト情報２１１３として生成されている。

【００７１】

以下、図２０及び図２１を用いて、差分抽出処理について具体的に説明する。本実施例では、図３のステップＳ１３４で説明したように、ポインタ情報は印刷時にイメージデータとして電子ファイルに付加されるため、ステップＳ１２５で検索処理によって電子ファイルを入手した段階では、まだ電子ファイルにポインタ情報は付加されていない。そこで、まず、電子ファイル２１１０に対してポインタ情報である２次元バーコード２１１４を付加する（ステップＳ２００１）。

【００７２】

次に、イメージデータとして付加されたポインタ情報である２次元バーコード２１１４を含む電子ファイル２１１０に対して、ブロックセレクション処理を行う（ステップＳ２００２）。ブロックセレクション処理に関しては、図４で説明を行った内容と同様である。図４では、入力画像（イメージデータ）に対して、入力画像中の各オブジェクト毎を塊として認識し、当該ブロックの各々を文字／図画／写真／線／表等の属性に判定し、異なる属性を持つ領域に分割する処理を説明したが、ステップＳ２００２では、これと同様の処理を電子ファイル２１１０に対して行うことになる。

【００７３】

そして、電子ファイル２１１０に対してブロックセレクション処理をされた後に、各々のオブジェクトに対して既知のＯＣＲ／ＯＭＲ処理を行う（ステップＳ２００３）。尚、当該処理は、図３のステップＳ１２２と同様の処理である。

【００７４】

次いで、電子ファイル２１１０から、ステップＳ２００１で付加したポインタ情報である２次元バーコード２１１４を検出する（ステップＳ２００４）。尚、当該処理は、図３のステップＳ１２３と同様の処理である。

【００７５】

さらに、図３のステップＳ１２３において検出済みの入力画像２１１２のブロックレイアウト情報２１１３から得られるポインタ情報である２次元バーコード２１１７の座標と、ステップＳ２００４で検出した電子ファイル２１１０のブロックレイアウト情報２１１１から得られるポインタ情報である２次元バーコード２１１５の座標とを比較して、入力画像２１１２と電子ファイル２１１０の向き、大きさ等を比較する（ステップＳ２００５）。

【００７６】

そして、ステップＳ２００５での比較結果から、入力画像２１１２に対する回転、拡大縮小、傾き補正等の補正処理を必要に応じて行う（ステップＳ２００６）。これらの処理は、画像に対してだけ行うのではなく、ブロックセレクション後のブロックレイアウト情報２１１１、２１１３やブロック情報、さらにポインタ情報である２次元バーコードブロック２１１５、２１１７にも施す。

【００７７】

また、ブロックレイアウト情報、ブロック情報及びポインタ情報に関しては、回転、拡大縮小、傾き補正等の補正処理を行った入力画像２１１２に対して、一連のブロックセレクション処理、ＯＣＲ／ＯＭＲ処理、ポインタ情報検出処理を再度行って更新するようにしてもよい。

【００７８】

いずれにしても、ステップＳ２００６によって、入力画像２１１２は、電子ファイル２１１０と向きや大きさが揃うことになる。尚、図２１に示す例では、同じ向き、等倍を示しているため、これらの処理を行う必要がない。

【００７９】

次いで、入力画像２１１２と電子ファイル２１１０の各々ブロックセレクション処理結果のブロックレイアウト情報２１１３、２１１１において、分割されたブロックごとに同一内容であるか否かの同一性を判断する（ステップＳ２００７）。この判断は、各々のブロックレイアウト情報、ブロック情報、及びＯＣＲ／ＯＭＲ情報から行う。図２１では、手書き文字２１１８部分のブロック２１１９の存在によって、入力画像２１１２が電子ファイル２１１０と異なるブロックを有すると判断される。

【００８０】

そして、ステップＳ２００７で同一と判断されたブロックを入力画像から消去する（ステップＳ２００８）。これによって、例えば図２１では、手書き文字ブロック２１１９以外のブロックが消去される。

【００８１】

さらに、ステップＳ２００８で消去されずに残ったブロック（本実施例では、ブロック２１１９）を、入力画像２１１２と電子ファイル２１１０間の差分イメージデータとして抽出する。図２１では手、書き文字ブロック２１１９が差分イメージデータとして差分抽出される。

【００８２】

［ベクトル化処理］

次に、図３のステップＳ１３２で示されるイメージ全体のベクトル化について詳細に説明する。本実施例では、上述した差分抽出によって抽出された差分情報のみをベクトル化する場合と、ステップＳ１２８で電子ファイルが特定できなかった場合にイメージデータ全体をベクトル化する処理との２通りが考えられるが、ここでは、まず、後者（イメージ全体のベクトル化）について説明する。そこで、文字ブロックに対しては各文字に対して文字認識処理を行う。

【００８３】

《文字認識処理》

文字認識処理では、入力されたイメージデータから文字単位で切り出された画像に対し、パターンマッチングの一手法を用いて認識を行い、対応する文字コードを得る。この認識処理は、文字画像から得られる特徴を数十次元の数値列に変換した観測特徴ベクトルと、あらかじめ字種毎に求められている辞書特徴ベクトルと比較し、最も距離の近い字種を認識結果とする処理である。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とする方法がある。

【００８４】

ステップＳ１２１のブロックセレクション処理で抽出された文字領域に対して文字認識を行う場合は、まず、該当領域に対して横書き、縦書きの判定を行う。次いで、各々対応する方向に行を切り出し、その後文字を切り出して文字画像を得る。横書き、縦書きの判定は、該当領域内で画素値に対する水平／垂直の射影を取り、水平射影の分散が大きい場合は横書き領域、垂直射影の分散が大きい場合は縦書き領域と判断すればよい。文字列及び文字への分解は、横書きの場合は水平方向の射影を利用して行を切り出し、さらに切り出された行に対する垂直方向の射影から、文字を切り出すことで行う。一方、縦書きの文字領域に対しては、水平と垂直を逆にすればよい。尚、この時文字のサイズを検出することができる。

【００８５】

《フォント認識処理》

文字認識の際に用いられる字種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、マッチングの際に文字コードとともにフォント種を出力する

ことで、文字のフォントを認識することができる。

【００８６】

《文字のベクトル化》

前述した文字認識処理及びフォント認識処理によって得られた文字コード及びフォント情報を用いて、各々あらかじめ用意されたアウトラインデータを用いて、文字部分の情報をベクトルデータに変換する。尚、入力された原稿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ベクトルデータとともに記録する。

【００８７】

以上の処理により、文字ブロックに属するイメージ情報をほぼ形状、大きさ、色が忠実なベクトルデータに変換することができる。

【００８８】

《文字以外の部分のベクトル化》

ステップＳ１２１のブロックセレクション処理で、図画或いは線、表領域とされた領域を対象として、それぞれ抽出された画素塊の輪郭をベクトルデータに変換する。具体的には、輪郭を成す画素の点列を角とみなされる点で区切って、各区間を部分的な直線或いは曲線で近似する。ここで、「角」とは、曲率が極大となる点である。

【００８９】

図１１は、曲率が極大となる点を説明するための図である。図１１に示すように、任意点Ｐiに対して左右ｋ個の離れた点Ｐi-k～Ｐi+kの間に弦を引いたとき、この弦とＰiの距離が極大となる点として求められる。さらに、Ｐi-k～Ｐi+k間の弦の長さ／弧の長さをＲとし、Ｒの値が閾値以下である点を角とみなすことができる。角によって分割された後の各区間は、直線は点列に対する最小二乗法等を用いて、曲線は３次スプライン関数等を用いてベクトル化することができる。

【００９０】

また、対象が内輪郭を持つ場合、ブロックセレクション処理で抽出した白画素輪郭の点列を用いて、同様に部分的直線或いは曲線で近似する。

【００９１】

以上のように、輪郭の区分線近似を用いることによって、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。尚、入力される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ベクトルデータとともに記録する。

【００９２】

図１２は、外輪郭が内輪郭又は別の外輪郭と近接している場合に太さを持った線として表現する例について説明するための図である。図１２に示すように、ある区間で外輪郭が、内輪郭又は別の外輪郭が近接している場合、２つの輪郭線を一まとめにし、太さを持った線として表現することができる。具体的には、ある輪郭の各点Ｐiから別輪郭上で最短距離となる点Ｑiまで線を引き、各距離ＰＱiが平均的に一定長以下の場合、注目区間はＰＱi中点を点列として直線又は曲線で近似し、その太さはＰＱiの平均値とする。線や線の集合体である表罫線は、前記したような太さを持つ線の集合として、効率よくベクトル表現することができる。

【００９３】

尚、文字ブロックに対する文字認識処理を用いたベクトル化については前述したように、当該文字認識処理の結果、辞書からの距離が最も近い文字を認識結果として用いる。ここで、この距離が所定値以上の場合は、必ずしも本来の文字に一致するとは限らず、形状が類似する文字に誤認識するような場合が多い。従って、本実施例では、このような文字に対しては上記したように、一般的な線画と同様に扱って当該文字をアウトライン化する。すなわち、従来は文字認識処理で誤認識を起こしていたような文字でも、誤った文字にベクトル化されることなく、可視的にイメージデータに忠実なアウトライン化によるベクトル化を行うことができる。また、写真と判定されたブロックに対しては、本発明ではベクトル化せずに、イメージデータのままとする。

【００９４】

《図形認識》
ここでは、上述したように任意形状の図形のアウトラインをベクトル化した後、これらのベクトル化された区分線を図形オブジェクト毎にグループ化する処理について説明する。

【００９５】
図１３は、ベクトルデータを図形オブジェクト毎にグループ化するまでの処理手順を説明するためのフローチャートである。まず、各ベクトルデータの始点、終点を算出する（ステップＳ７００）。次に、各ベクトルの始点、終点情報を用いて、図形要素を検出する（ステップＳ７０１）。ここで、図形要素の検出とは、区分線が構成している閉図形を検出することである。検出に際しては、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して検出を行う。

【００９６】
次に、図形要素内に存在する他の図形要素又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（ステップＳ７０２）。尚、図形要素内に他の図形要素又は区分線が存在しない場合は、図形要素を図形オブジェクトとする。

【００９７】
図１４は、図形要素を検出する処理手順を説明するためのフローチャートである。まず、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、閉図形構成ベクトルを抽出する（ステップＳ７１０）。次に、閉図形構成ベクトルの中から当該ベクトルの始点を開始点とし、時計回りに順にベクトルを追っていく。そして、開始点に戻るまで追跡を行い、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（ステップＳ７１１）。尚、この際に、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。最後に、ステップＳ７１０で除去された不要ベクトルのうち、ステップＳ７１１で閉図形としてグループ化されたベクトルに接合しているものを検出し、一つの図形要素としてグループ化する（ステップＳ７１２）。

【００９８】
以上の処理によって、図形ブロックを個別に再利用可能な個別の図形オブジェクトとして扱うことが可能になる。

【００９９】
［差分情報のベクトル化処理］
次に、ステップＳ１３２のベクトル化処理のうち、前述の前者である図２０で抽出された差分情報のベクトル化処理について説明する。

【０１００】
ベクトル化に関しては、上述した［ベクトル化処理］と同等な処理を行う。これにより、手書きされた文字、例えば図２１で示した手書き文字２１１８は、文字認識された後、文字のベクトル化又はフォント化される。

【０１０１】
ここで、手書き文字については、ステップＳ１２１のブロックセレクション処理の結果、場合によってはテキストブロックと認識されずに細線と認識され、ラインブロックと判断される可能性がある。そこで、このような場合には、前述した［ベクトル化処理］内の項目《文字以外の部分のベクトル化》で説明したように、輪郭を検出してアウトライン化することで、可視的にイメージデータに忠実なアウトラインによるベクトル化を行うことができる。

【０１０２】
図２１では手書き文字２１１８での例を示しているが、手書きの細線や図面等に関しても同様であり、例えば細線等はアウトラインによるベクトル化、図面や図形等に関しては、前述した《図形認識》と同様な処理を行うことでベクトル化を行うことができる。

【０１０３】
［電子ファイルとの合成］

次に、入力画像と電子ファイルとの差分部分のベクトル化が終了し、オリジナルの電子ファイルに当該差分部分のベクトルデータを合成する処理について説明する。図２２は、図２１の入力画像データ２１１２の手書き文字２１１８をベクトル化して合成した電子ファイルの一例を示す図である。図２２において、２２００は、図２１の電子ファイル２１１０に手書き文字２１１８をベクトル化したベクトル文字２２０１を付加した合成後の新たな電子ファイルデータを示す。

【０１０４】

図２２に示すように、ベクトル化された手書き文字２１１８は、ブロックレイアウト情報２１１３の座標に手書き文字に最も似たフォント及びサイズで、ベクトル文字２２０１として元の電子ファイル２１１０に合成される。

【０１０５】

以上説明したように、入力画像中に手書き文字やオリジナルの電子ファイルと異なる情報が含まれているような場合であっても、入力画像と電子ファイルとの差分情報の抽出を行い、その差分情報のベクトル化を行い、元の電子ファイルに合成し、保存や格納アドレスの通知を行うことで、入力されたイメージデータを新たな電子ファイルとして取得、或いは電子ファイルの更新を好適に行うことができる。

【０１０６】

［アプリデータへの変換処理］

図１５は、一頁分のイメージデータをブロックセレクション処理（ステップＳ１２１）及びベクトル化処理（ステップＳ１３２）によって変換された結果として得られる中間データ形式のファイルのデータ構造を示す図である。図１５に示すようなデータ形式は、ドキュメント・アナリシス・アウトプット・フォーマット（ＤＡＯＦ）と呼ばれる。すなわち、図１５は、ＤＡＯＦのデータ構造を示す図である。

【０１０７】

図１５において、７９１はHeader（ヘッダ）であり、処理対象の文書画像データに関する情報が保持される。７９２はレイアウト記述データ部であり、文書画像データ中のTEXT（文字）、TITLE（タイトル）、CAPTION（キャプション）、LINEART（線画）、PICTURE（自然画）、FRAME（枠）、TABLE（表）等の属性毎に認識された各ブロックの属性情報とその矩形アドレス情報を保持する。

【０１０８】

７９３は文字認識記述データ部であり、TEXT、TITLE、CAPTION等のTEXTブロックを文字認識して得られる文字認識結果を保持する。７９４は表記述データ部であり、TABLEブロックの構造の詳細を格納する。７９５は画像記述データ部であり、PICTUREやLINEART等のブロックのイメージデータを文書画像データから切り出して保持する。

【０１０９】

このようなＤＡＯＦは、中間データとしてのみならず、それ自体がファイル化されて保存される場合もあるが、このファイルの状態では、一般の文書作成アプリケーションで個々のオブジェクトを再利用することはできない。そこで、次に、ＤＡＯＦからアプリデータに変換する処理（ステップＳ１３０）について詳説する。

【０１１０】

図１６は、アプリデータへの変換処理全体の概略手順を説明するためのフローチャートである。まず、ＤＡＯＦデータを入力する（ステップＳ８００）。次いで、アプリデータの元となる文書構造ツリー生成を行う（ステップＳ８０２）。そして、生成した文書構造ツリーに基づいて、ＤＡＯＦ内の実データを流し込み、実際のアプリデータを生成する（ステップＳ８０４）。

【０１１１】

図１７は、文書構造ツリー生成処理（ステップＳ８０２）の詳細な処理手順を説明するためのフローチャートである。また、図１８は、文書構造ツリーの概要を説明するための図である。尚、全体制御の基本ルールとして、処理の流れはミクロブロック（単一ブロック）からマクロブロック（ブロックの集合体）へ移行するものとする。また、以後の説明

では、ブロックとは、ミクロブロック及びマクロブロック全体を指す。

【０１１２】

まず、ブロック単位で縦方向の関連性を元に再グループ化する（ステップＳ８０２ａ）。尚、スタート直後はミクロブロック単位での判定となる。ここで、関連性とは、距離が近く、ブロック幅（横方向の場合は高さ）がほぼ同一であること等で定義することができる。また、距離、幅、高さ等の情報はＤＡＯＦを参照して抽出する。

【０１１３】

図１８において、（ａ）は実際のページ構成、（ｂ）はその文書構造ツリーを示している。ステップＳ８０２ａのグループ化の結果、Ｔ３、Ｔ４、Ｔ５が一つのグループＶ１として、Ｔ６、Ｔ７が一つのグループＶ２として、それぞれ同じ階層のグループとして生成される。

【０１１４】

次に、縦方向のセパレータの有無をチェックする（ステップＳ８０２ｂ）。セパレータは、例えば、物理的にはＤＡＯＦ中でライン属性を持つオブジェクトである。また、論理的な意味としては、アプリ中で明示的にブロックを分割する要素である。ここでセパレータを検出した場合は、同じ階層で再分割する。

【０１１５】

次いで、分割がこれ以上存在し得ないか否かをグループ長を利用して判定する（ステップＳ８０２ｃ）。例えば、縦方向のグルーピング長がページ高さか否かを判定する。その結果、縦方向のグループ長がページ高さとなっている場合は（Ｙｅｓ）、文書構造ツリー生成は終了する。例えば、図１８に示すような構造の場合は、セパレータもなく、グループ高さはページ高さではないので、Ｎｏと判定され、ステップＳ８０２ｄに進む。

【０１１６】

ステップＳ８０２ｄでは、ブロック単位で横方向の関連性を元に再グループ化する。但し、この再グループ化においてもスタート直後の第一回目は、ミクロブロック単位で判定を行うことになる。また、関連性及びその判定情報の定義は、縦方向の場合と同じである。例えば、図１８の構造の場合は、Ｔ１とＴ２でＨ１、Ｖ１とＶ２でＨ２が生成され、Ｈ１はＴ１、Ｔ２の一つ上、Ｈ２はＶ１、Ｖ２の１つ上の同じ階層のグループとして生成される。

【０１１７】

次いで、横方向セパレータの有無をチェックする（ステップＳ８０２ｅ）。図１８では、Ｓ１があるので、これをツリーに登録し、Ｈ１、Ｓ１、Ｈ２という階層が生成される。そして、分割がこれ以上存在し得ないか否かをグループ長を利用して判定する（ステップＳ８０２ｆ）。例えば、横方向のグルーピング長がページ幅か否かを判定する。その結果、横方向のグループ長がページ幅となっている場合（Ｙｅｓ）、文書構造ツリー生成は終了する。一方、ページ幅となっていない場合（Ｎｏ）は、ステップＳ８０２ｂに戻り、再度もう一段上の階層で、縦方向の関連性チェックから繰り返す。例えば、図１８の構造の場合は、分割幅がページ幅になっているので、ここで終了し、最後にページ全体を表す最上位階層のＶ０が文書構造ツリーに付加される。

【０１１８】

文書構造ツリーが完成した後、その情報に基づいて、ステップＳ８０４においてアプリデータの生成を行う。図１８の構造の場合は、具体的に以下のようになる。

【０１１９】

すなわち、Ｈ１は横方向に２つのブロックＴ１、Ｔ２があるので、２カラムとし、Ｔ１の内部情報（ＤＡＯＦを参照した文字認識結果の文章や画像等）を出力後、カラムを変えて、Ｔ２の内部情報出力し、その後Ｓ１を出力する。また、Ｈ２は横方向に２つのブロックＶ１、Ｖ２があるので、２カラムとして出力し、Ｖ１はＴ３、Ｔ４、Ｔ５の順にその内部情報を出力し、その後カラムを変えて、Ｖ２のＴ６、Ｔ７の内部情報を出力する。以上により、アプリデータへの変換処理を行うことができる。

【０１２０】

［ポインタ情報の付加］

次に、ステップＳ１３４で示されるポインタ情報付加処理について詳細に説明する。処理すべき文書が検索処理で特定された場合、或いはベクトル化によって元ファイルが再生できた場合において、該文書を記録処理する場合においては、紙への記録の際にポインタ情報を付与することで、この文書を用いて再度各種処理を行う場合に簡単に元ファイルデータを取得できる。

【０１２１】

図１９は、ポインタ情報としてのデータ文字列を２次元バーコード（ＱＲコードシンボル：ＪＩＳ　Ｘ０５１０）３１１を用いて符号化して画像中に付加する手順を説明するためのフローチャートである。

【０１２２】

２次元バーコード内に組み込むデータは、対応するファイルが格納されるサーバアドレス情報を表しており、例えばファイルサーバ名からなるパス情報で構成される。或いは、対応するサーバのＵＲＬや、対応するファイルが格納されているデータベース１０５ａ、ｂ内或いはＭＦＰ１００自体が有する記憶装置１１１を管理するためのＩＤ等で構成される。

【０１２３】

まず、符号化する種種の異なる文字を識別するため、入力データ列を分析する。また、誤り検出及び誤り訂正レベルを選択し、入力データが収容できる最小型番を選択する（ステップＳ９００）。次に、入力データ列を所定のビット列に変換し、必要に応じてデータのモード（数字、英数字、８ビットバイト、漢字等)を表す指示子や、終端パターンを付加する。さらに、所定のビットコード語に変換することによってデータの符号化を行う（ステップＳ９０１）。

【０１２４】

この時、誤り訂正を行うため、コード語列を型番及び誤り訂正レベルに応じて所定のブロック数に分割し、各ブロック毎に誤り訂正コード語を生成し、データコード語列の後に付加する（ステップＳ９０２）。さらに、ステップＳ９０２で得られた各ブロックのデータコード語を接続し、各ブロックの誤り訂正コード語、また必要に応じて剰余コード語を接続して、メッセージの構築を行う（ステップＳ９０３）。

【０１２５】

次に、マトリクスに位置検出パターン、分離パターン、タイミングパターン及び位置合わせパターン等とともにコード語モジュールを配置する（ステップＳ９０４）。さらに、シンボルの符号化領域に対して最適なマスクパターンを選択して、マスク処理パターンをステップＳ９０４で得られたモジュールにＸＯＲ演算により変換する（ステップＳ９０５）。最後に、ステップＳ９０５で得られたモジュールに形式情報及び型番情報を生成して、２次元コードシンボルを完成させる（ステップＳ９０６）。

【０１２６】

上述したサーバアドレス情報の組み込まれた２次元バーコードは、例えば、クライアントＰＣ１０２から電子ファイルをプリントデータとして形成装置１１２で紙上に記録画像として形成する場合に、データ処理装置１１５内で記録可能なラスタデータに変換された後にラスタデータ上の所定の個所に付加されて画像形成される。ここで、画像形成された紙を配布されたユーザは、画像読み取り部１１０で読み取ることにより、前述したステップＳ１２３においてポインタ情報からオリジナルの電子ファイルが格納されているサーバの場所を適切に検出することができる。

【０１２７】

尚、同様の目的で付加情報を付与する手段は、本実施例で説明した２次元バーコードの他に、例えば、ポインタ情報を直接文字列で文書に付加する方法、文書内の文字列、特に文字と文字の間隔を変調して情報を埋め込む方法、文書中の中間調画像中に埋め込む方法等の一般に電子透かしと呼ばれる方法を適用してもよい。

【０１２８】

上述したように本実施例によれば、ユーザが加筆した手書き文書情報等を含めたあらゆる紙文書上の情報を消失することなく再利用可能な電子ファイルとして保存することができる。また、それらの情報をベクトルデータとすることで、データ量を減少することができ、入力された原稿の再利用性、編集可能性を大きく高めることもできる。

【０１２９】

また、本実施例によれば、原稿を読み取り走査することによって得られたイメージ情報から、当該原稿のオリジナルの電子ファイルを特定し、上記イメージ情報に一致するオリジナルの電子ファイルを特定することで、入力イメージ情報とオリジナル電子ファイルとの差分情報を抽出することができ、オリジナルには存在しない追記情報を取り扱うことが可能になる。

【０１３０】

また、ベクトル化された差分情報をオリジナルの電子ファイルに合成し、オリジナルの電子ファイルに上書き保存する。或いは、ベクトル化された差分情報とオリジナルの電子ファイルとを合成して新たな電子ファイルとして生成する。或いは、ベクトル化された差分情報をオリジナルの電子ファイル内の別レイヤーとして保存する。或いは、ベクトル化された差分情報をオリジナルの電子ファイルとは別の差分情報ファイルとして保存する等、ユーザが選択を行うことにより、オリジナルの電子データと差分情報データの生成、保存に対して、ファイル容量を削減したり、検索を高速化したり、オリジナル性の確保を重視したデータ処理が可能となる。

【実施例２】

【０１３１】

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。

【０１３２】

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。

【０１３３】

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。

【０１３４】

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。

【０１３５】

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。

【０１３６】

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。

【０１３７】

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。

【０１３８】

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。

【０１３９】

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】

【０１４０】

【図１】本発明の第１の実施例に係る画像処理システムの構成を示すブロック図である。

【図２】本発明の第１の実施例に係るＭＦＰ１００の構成を示すブロック図である。

【図３】本発明の第１の実施例に係る画像処理システムによる画像処理の手順について説明するためのフローチャートである。

【図４】ブロックセレクション処理によって読み取った１枚のイメージデータを属性を判定し複数のブロックに分割する様子を示す図である。

【図５】ブロックセレクション処理で得られた各ブロックに対するブロック情報の一例について示す図である。

【図６】原稿画像中に付加された２次元バーコード（ＱＲコードシンボル）を復号してデータ文字列を出力する手順を説明するためのフローチャートである。

【図７】２次元バーコードが付加された原稿３１０の一例を示す図である。

【図８】検出されたポインタ情報から電子ファイルを検索する処理手順について説明するためのフローチャートである。

【図９】ファイルアクセス権を含むポインタ情報から電子ファイルが格納されているサーバを検索する処理手順について説明するためのフローチャートである。

【図１０】データベース内から入力イメージファイルに類似した電子ファイルのレイアウト検索を行う手順について説明するためのフローチャートである。

【図１１】曲率が極大となる点を説明するための図である。

【図１２】外輪郭が内輪郭又は別の外輪郭と近接している場合に太さを持った線として表現する例について説明するための図である。

【図１３】ベクトルデータを図形オブジェクト毎にグループ化するまでの処理手順を説明するためのフローチャートである。

【図１４】図形要素を検出する処理手順を説明するためのフローチャートである。

【図１５】一頁分のイメージデータをブロックセレクション処理及びベクトル化処理によって変換された結果として得られる中間データ形式のファイルのデータ構造を示す図である。

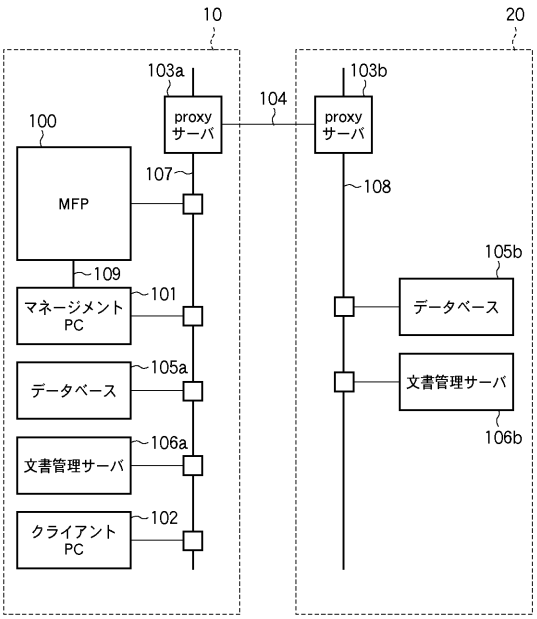
【図１６】アプリデータへの変換処理全体の概略手順を説明するためのフローチャートである。

【図１７】文書構造ツリー生成処理の詳細な処理手順を説明するためのフローチャートである。

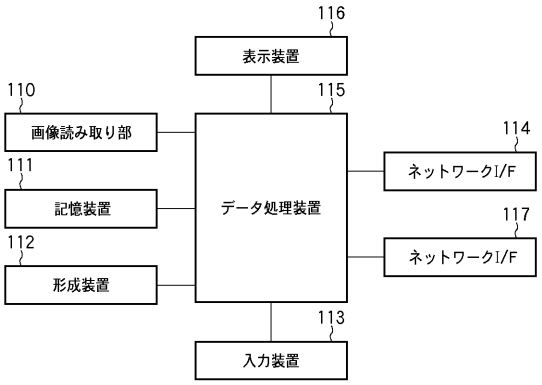
【図１８】文書構造ツリーの概要を説明するための図である。

【図１９】ポインタ情報としてのデータ文字列を２次元バーコードを用いて符号化して画像中に付加する手順を説明するためのフローチャートである。
【図２０】図３に示すフローチャートにおける差分抽出処理（ステップＳ１３１）の手順を説明するための概略を示すフローチャートである。
【図２１】図２０に示す差分抽出処理内容を説明するための電子ファイルに関する入力画像のイメージデータ例及びブロックセレクション例を示す図である。
【図２２】図２１の入力画像データ２１１２の手書き文字２１１８をベクトル化して合成した電子ファイルの一例を示す図である。
【符号の説明】
【０１４１】
１００　デジタル複合機（ＭＦＰ）
１０１　マネージメントＰＣ
１０２　クライアントＰＣ
１０３ａ、１０３ｂ　プロキシサーバ
１０４　ネットワーク
１０５ａ、１０５ｂ　データベース
１０６ａ、１０６ｂ　文書管理サーバ
１０７、１０８、１０９　ＬＡＮ
１１０　画像読み取り部
１１１　記憶装置
１１２　形成装置
１１３　入力装置
１１４、１１７　ネットワークＩ／Ｆ
１１５　データ処理装置
１１６　表示装置

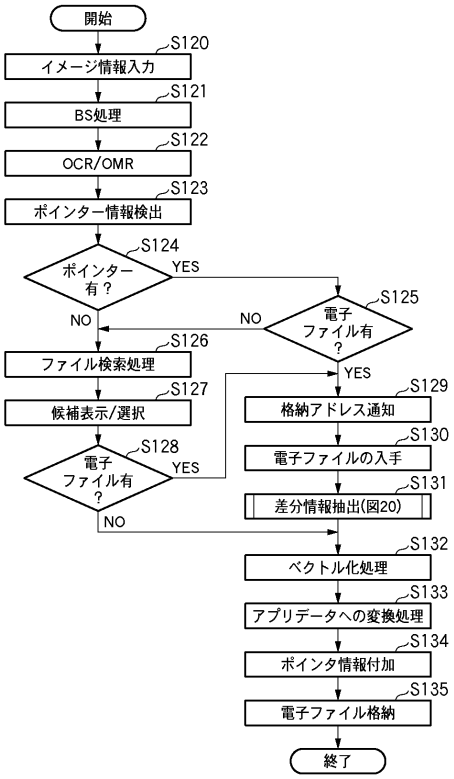
【図１】



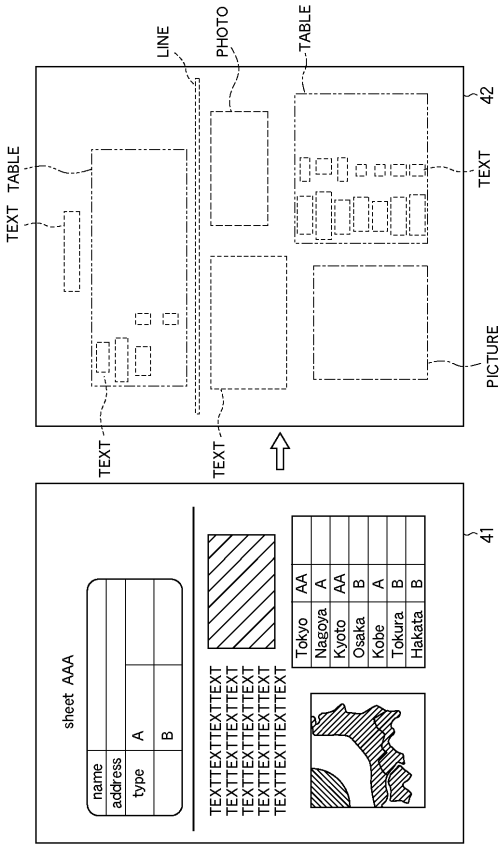

【図２】



【図３】




【図４】




【図５】

ブロック情報

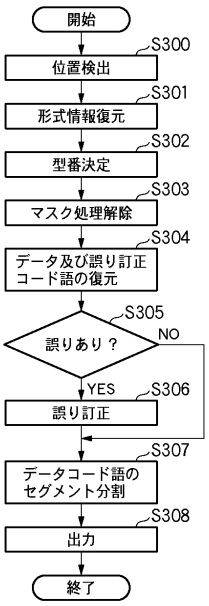
| | 属性 | 座標X | 座標Y | 幅W | 高さH | OCR情報 |
|---|---|---|---|---|---|---|
| ブロック1 | 1 | X1 | Y1 | W1 | H1 | 有 |
| ブロック2 | 3 | X2 | Y2 | W2 | H2 | 有 |
| ブロック3 | 2 | X3 | Y3 | W3 | H3 | 無 |
| ブロック4 | 1 | X4 | Y4 | W4 | H4 | 有 |
| ブロック5 | 3 | X5 | Y5 | W5 | H5 | 有 |
| ブロック6 | 5 | X6 | Y6 | W6 | H6 | 無 |

属性 1 : text 2 : picture 3 : table 4 : line 5 : photo

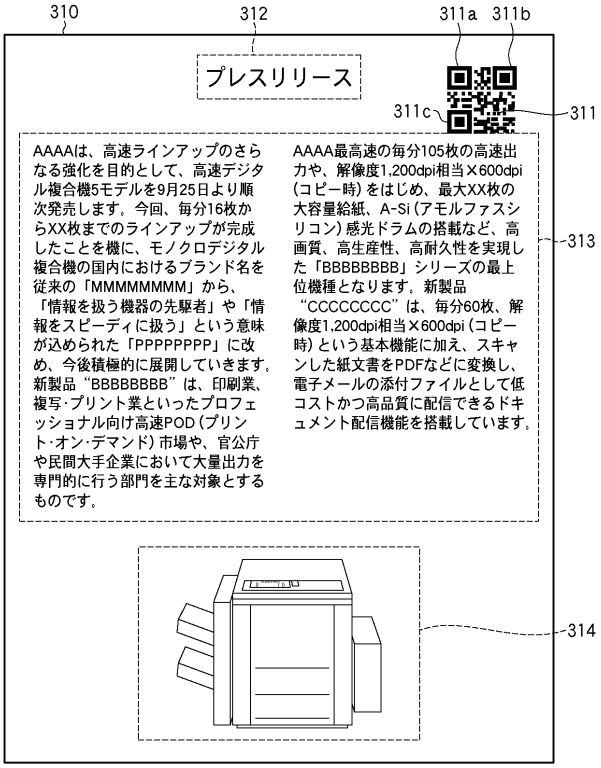
入力ファイル情報

| | |
|---|---|
| ブロック総数 | N (=6) |

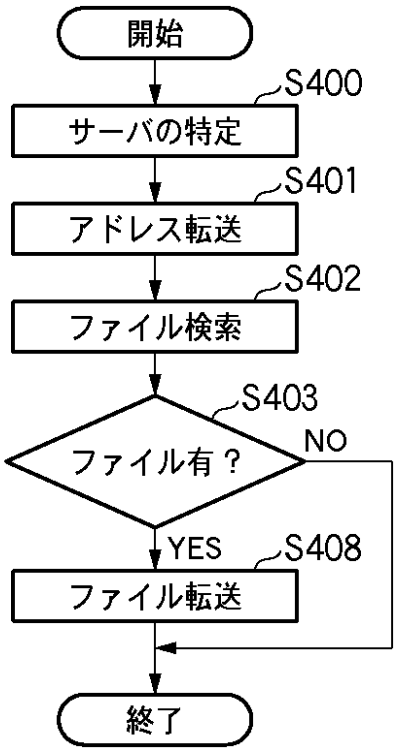
【図６】



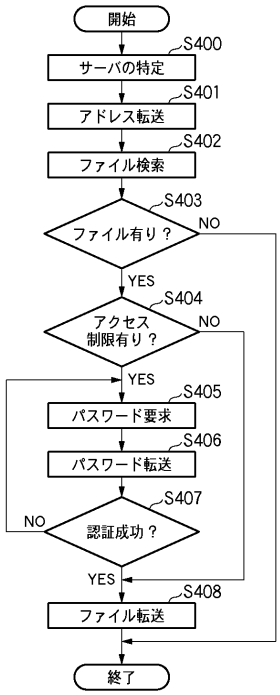
【図７】


310
312
311a
311b
プレスリリース
311c
311
AAAAは、高速ラインアップのさらなる強化を目的として、高速デジタル複合機5モデルを9月25日より順次発売します。今回、毎分16枚からXX枚までのラインアップが完成したことを機に、モノクロデジタル複合機の国内におけるブランド名を従来の「MMMMMMMM」から、「情報を扱う機器の先駆者」や「情報をスピーディに扱う」という意味が込められた「PPPPPPPP」に改め、今後積極的に展開していきます。新製品“BBBBBBBB”は、印刷業、複写・プリント業といったプロフェッショナル向け高速POD（プリント・オン・デマンド）市場や、官公庁や民間大手企業において大量出力を専門的に行う部門を主な対象とするものです。
AAAA最高速の毎分105枚の高速出力や、解像度1,200dpi相当×600dpi（コピー時）をはじめ、最大XX枚の大容量給紙、A-Si（アモルファスシリコン）感光ドラムの搭載など、高画質、高生産性、高耐久性を実現した「BBBBBBBB」シリーズの最上位機種となります。新製品“CCCCCCCC”は、毎分60枚、解像度1,200dpi相当×600dpi（コピー時）という基本機能に加え、スキャンした紙文書をPDFなどに変換し、電子メールの添付ファイルとして低コストかつ高品質に配信できるドキュメント配信機能を搭載しています。
313
314

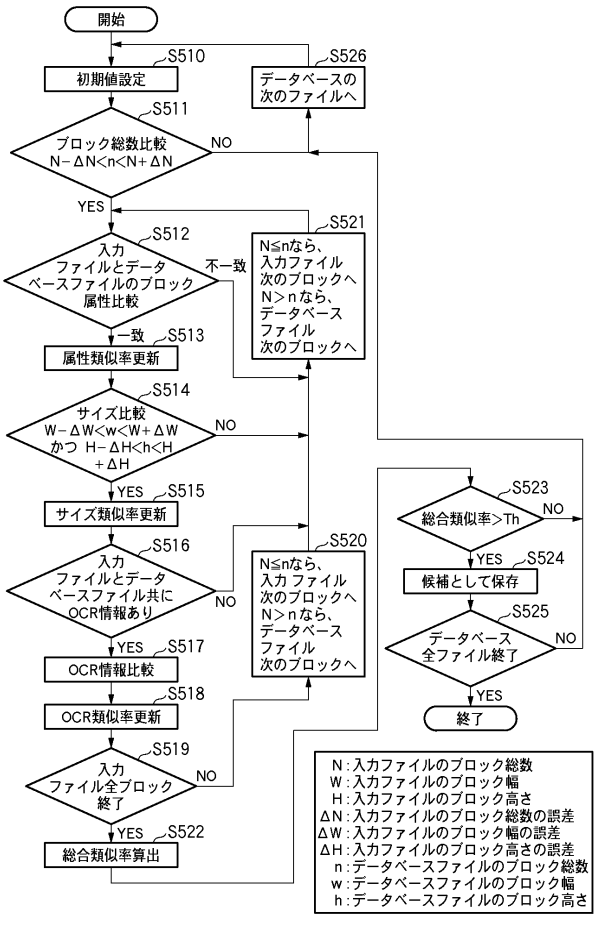
【図８】


開始
S400
サーバの特定
S401
アドレス転送
S402
ファイル検索
S403
ファイル有？
NO
YES
S408
ファイル転送
終了

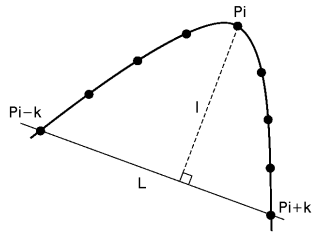
【図９】


開始
S400
サーバの特定
S401
アドレス転送
S402
ファイル検索
S403
ファイル有り？
NO
YES
S404
アクセス制限有り？
NO
YES
S405
パスワード要求
S406
パスワード転送
S407
認証成功？
NO
YES
S408
ファイル転送
終了

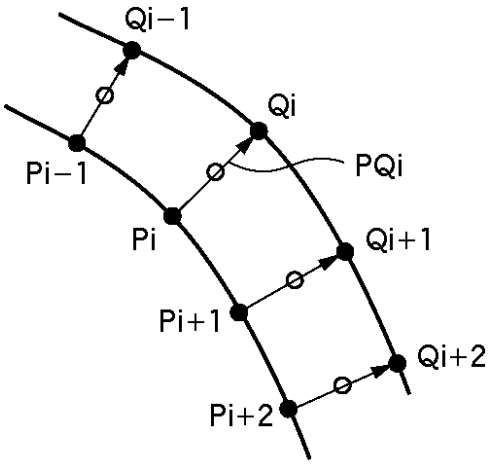
【図１０】


開始
S510
初期値設定
S511
ブロック総数比較
N−ΔN<n<N+ΔN
NO
YES
S512
入力ファイルとデータベースファイルのブロック属性比較
不一致
一致
S513
属性類似率更新
S514
サイズ比較
W−ΔW<w<W+ΔW
かつ H−ΔH<h<H+ΔH
NO
YES
S515
サイズ類似率更新
S516
入力ファイルとデータベースファイル共にOCR情報あり
NO
YES
S517
OCR情報比較
S518
OCR類似率更新
S519
入力ファイル全ブロック終了
NO
YES
S522
総合類似率算出
S520
N≦nなら、入力ファイル次のブロックへ
N＞nなら、データベースファイル次のブロックへ
S521
N≦nなら、入力ファイル次のブロックへ
N＞nなら、データベースファイル次のブロックへ
S526
データベースの次のファイルへ
S523
総合類似率>Th
NO
YES
S524
候補として保存
S525
データベース全ファイル終了
NO
YES
終了
N：入力ファイルのブロック総数
W：入力ファイルのブロック幅
H：入力ファイルのブロック高さ
ΔN：入力ファイルのブロック総数の誤差
ΔW：入力ファイルのブロック幅の誤差
ΔH：入力ファイルのブロック高さの誤差
n：データベースファイルのブロック総数
w：データベースファイルのブロック幅
h：データベースファイルのブロック高さ

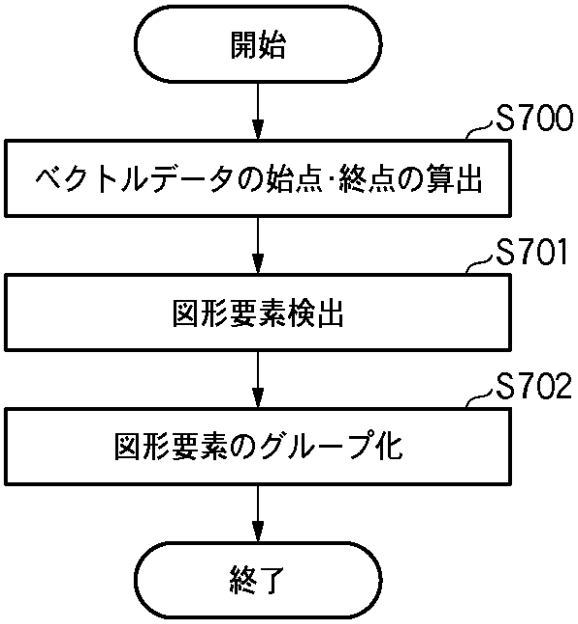
【図１１】



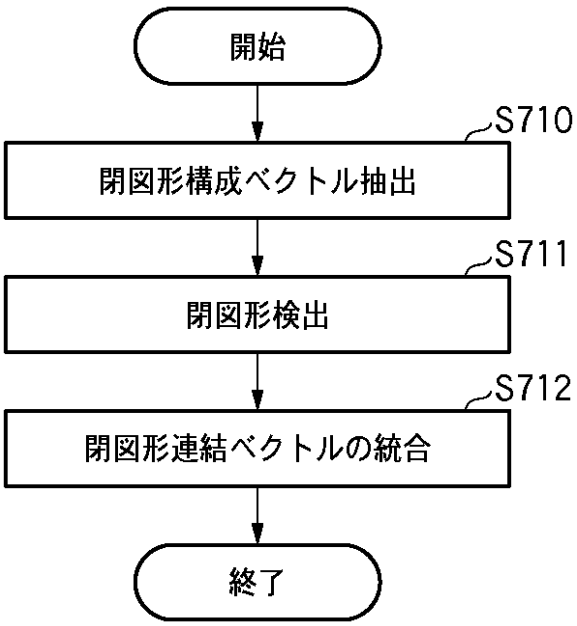

【図１２】



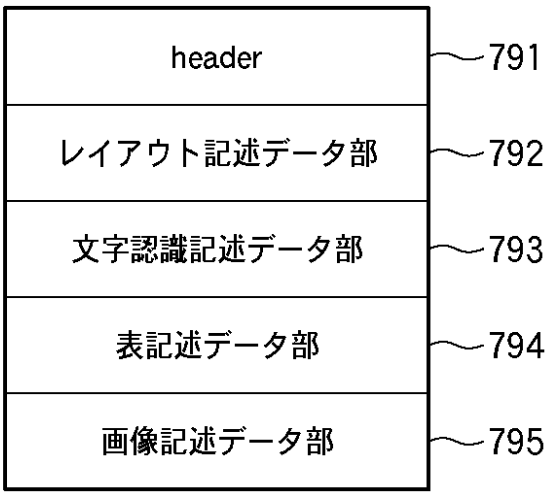

【図１３】



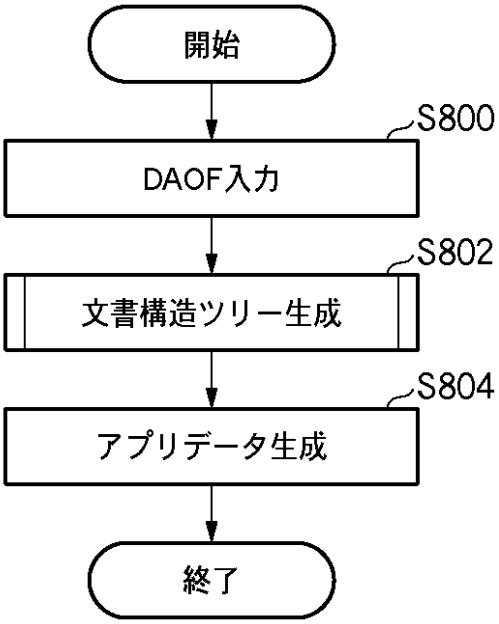

【図１４】



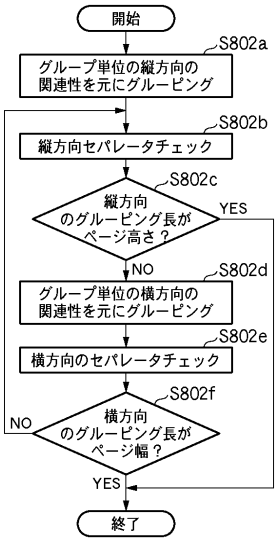

【図１５】



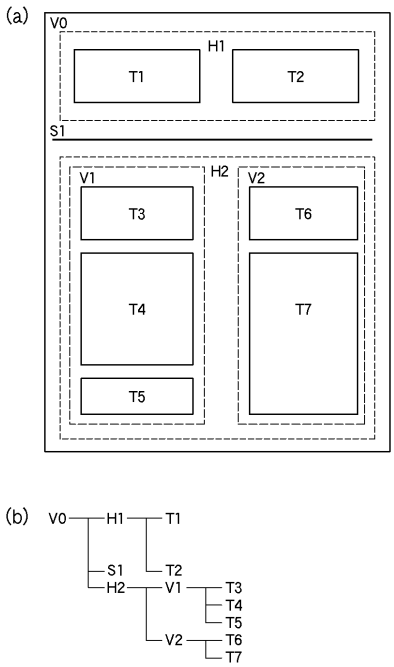
【図１６】



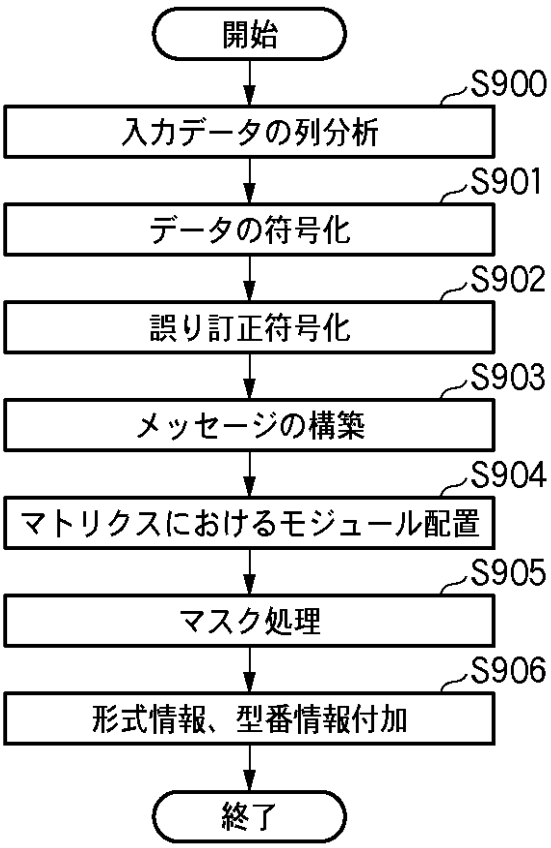

【図１７】



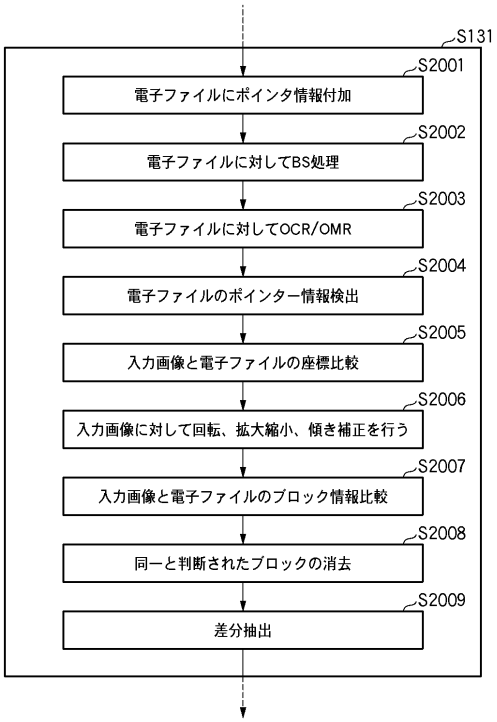

【図１８】



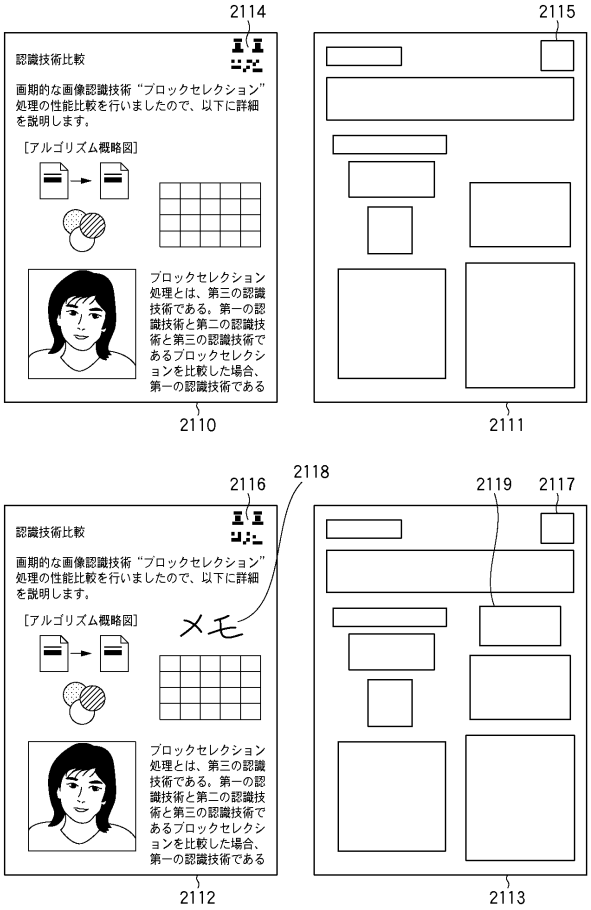

【図１９】



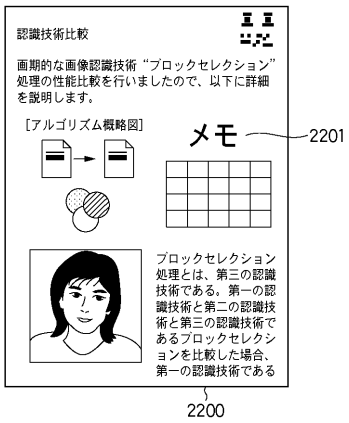
【図２０】


S131
S2001
電子ファイルにポインタ情報付加
S2002
電子ファイルに対してBS処理
S2003
電子ファイルに対してOCR/OMR
S2004
電子ファイルのポインター情報検出
S2005
入力画像と電子ファイルの座標比較
S2006
入力画像に対して回転、拡大縮小、傾き補正を行う
S2007
入力画像と電子ファイルのブロック情報比較
S2008
同一と判断されたブロックの消去
S2009
差分抽出

【図２１】


2114
2115
認識技術比較
画期的な画像認識技術“ブロックセレクション”処理の性能比較を行いましたので、以下に詳細を説明します。
[アルゴリズム概略図]
ブロックセレクション処理とは、第三の認識技術である。第一の認識技術と第二の認識技術と第三の認識技術であるブロックセレクションを比較した場合、第一の認識技術である
2110
2111
2116
2118
2119
2117
メモ
2112
2113

【図２２】


認識技術比較
画期的な画像認識技術“ブロックセレクション”処理の性能比較を行いましたので、以下に詳細を説明します。
[アルゴリズム概略図]
メモ
2201
ブロックセレクション処理とは、第三の認識技術である。第一の認識技術と第二の認識技術と第三の認識技術であるブロックセレクションを比較した場合、第一の認識技術である
2200

フロントページの続き

Ｆターム(参考) 5B050 AA10 BA10 BA16 BA18 BA20 EA06 EA18 GA08
5B075 KK24 ND07 PP04 PQ02 PQ32 QS03 UU33
5C073 AB02 AB07 AB12