(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第4590433号
(P4590433)

(45)発行日　平成22年12月1日(2010.12.1)　(24)登録日　平成22年9月17日(2010.9.17)

(51)Int.Cl.		F I		
G06K 9/00	(2006.01)	G06K	9/00	S
G06F 17/30	(2006.01)	G06F	17/30	230Z
		G06F	17/30	380Z

請求項の数 18　(全 25 頁)

(21)出願番号　特願2007-172737 (P2007-172737)
(22)出願日　平成19年6月29日 (2007.6.29)
(65)公開番号　特開2009-9527 (P2009-9527A)
(43)公開日　平成21年1月15日 (2009.1.15)
審査請求日　平成22年3月12日 (2010.3.12)

早期審査対象出願

(73)特許権者　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　100126240
弁理士　阿部　琢磨
(74)代理人　100124442
弁理士　黒岩　創吾
(72)発明者　金津　知俊
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　榎本　誠
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、コンピュータプログラム

(57)【特許請求の範囲】

【請求項１】

文書画像内の複数の文字画像に対して文字認識処理を行うことにより、それぞれの文字画像に対応する文字コードを得る文字認識手段と、

前記文書画像内の各文字画像についての文字の幅と文字行の高さとに基づいて、複数種類の字形データの中から、使用する字形データの種類を選択する選択手段と、

前記文書画像と、前記文字認識手段で得た複数の文字コードと、前記複数の文字コードに対応する文字を描画する際に複数の異なる文字コードで共通利用させるための前記複数種類の字形データと、前記複数の文字コードの描画の際に使用する字形データの種類を示すデータとを格納した電子文書を生成する生成手段と、を有し、

前記電子文書に格納される、前記複数の文字コードの描画の際に使用する字形データの種類を示すデータは、前記文書画像内の各文字画像に対して前記選択手段で選択された字形データの種類に基づいて記述されることを特徴とする画像処理装置。

【請求項２】

文書画像内の複数の文字画像に対して文字認識処理を行うことにより、それぞれの文字画像に対応する文字コードを得る文字認識手段と、

前記文書画像内の各文字画像についての文字送り幅と文字行の高さとに基づいて、複数種類の字形データの中から、使用する字形データの種類を選択する選択手段と、

前記文書画像と、前記文字認識手段で得た複数の文字コードと、前記複数の文字コードに対応する文字を描画する際に複数の異なる文字コードで共通利用させるための前記複数

種類の字形データと、前記複数の文字コードの描画の際に使用する字形データの種類を示すデータとを格納した電子文書を生成する生成手段と、を有し、
前記電子文書に格納される、前記複数の文字コードの描画の際に使用する字形データの種類を示すデータは、前記文書画像内の各文字画像に対して前記選択手段で選択された字形データの種類に基づいて記述されることを特徴とする画像処理装置。
【請求項３】
前記複数種類の字形データは、幅の異なる複数種類の字形データであることを特徴とする請求項１または２のいずれかに記載の画像処理装置。
【請求項４】
前記字形データは、矩形または三角または丸または直線のいずれかの形状を有する字形データであることを特徴とする請求項１乃至３のいずれかに記載の画像処理装置。
【請求項５】
前記字形データは、文字行の下部に描画を行う字形データであることを特徴とする請求項１乃至３のいずれかに記載の画像処理装置。
【請求項６】
前記生成手段は、文字間距離の平均値を更に格納させた前記電子文書を生成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】
前記生成手段で生成された電子文書には、前記複数の文字コードに対応させた字形データを、前記文書画像内の各文字画像に重なる位置に透明色で描画させるための記述が含まれることを特徴とする請求項１乃至６のいずれかに記載の画像処理装置。
【請求項８】
前記電子文書は、ＸＭＬフォーマットまたはＸＰＳフォーマットのいずれかで記述された電子文書であることを特徴とする請求項１乃至７のいずれかに記載の画像処理装置。
【請求項９】
前記画像処理装置は、前記文書画像を圧縮する圧縮手段を更に有し、
前記電子文書に格納される文書画像は、前記圧縮手段で圧縮処理が施された文書画像であることを特徴とする請求項１乃至８のいずれかに記載の画像処理装置。
【請求項１０】
前記圧縮手段は、前記文書画像内に含まれる領域を解析して適応的に圧縮することを特徴とする請求項９に記載の画像処理装置。
【請求項１１】
前記生成された電子文書に対して、入力されたキーワードで検索し、当該キーワードに一致する部分を強調表示させる検索手段を、更に有することを特徴とする請求項１乃至１０のいずれかに記載の画像処理装置。
【請求項１２】
前記検索手段は、前記キーワードに一致する部分の色を反転することにより強調表示させることを特徴とする請求項１１に記載の画像処理装置。
【請求項１３】
紙文書をスキャンして前記文書画像を生成するスキャナを更に有することを特徴とする請求項１乃至１２のいずれかに記載の画像処理装置。
【請求項１４】
文字認識手段が、文書画像内の複数の文字画像に対して文字認識処理を行うことにより、それぞれの文字画像に対応する文字コードを得る文字認識ステップと、
選択手段が、前記文書画像内の各文字画像についての文字の幅と文字行の高さとに基づいて、複数種類の字形データの中から、各文字コードに対応する文字を描画する際に使用する字形データの種類を選択する選択ステップと、
生成手段が、前記文書画像と、前記文字認識ステップで得た複数の文字コードと、前記複数の文字コードに対応する文字を描画する際に複数の異なる文字コードで共通利用させるための前記複数種類の字形データと、前記複数の文字コードの描画の際に使用する字形

データの種類を示すデータとを格納した電子文書を生成する生成ステップと、を有し、
前記電子文書に格納される、前記複数の文字コードの描画の際に使用する字形データの種類を示すデータは、前記文書画像内の各文字画像に対して前記選択ステップで選択された字形データの種類に基づいて記述されることを特徴とする画像処理方法。
【請求項１５】
文字認識手段が、文書画像内の複数の文字画像に対して文字認識処理を行うことにより、それぞれの文字画像に対応する文字コードを得る文字認識ステップと、
選択手段が、前記文書画像内の各文字画像についての文字送り幅と文字行の高さとに基づいて、複数種類の字形データの中から、使用する字形データの種類を選択する選択ステップと、
生成手段が、前記文書画像と、前記文字認識ステップで得た複数の文字コードと、前記複数の文字コードに対応する文字を描画する際に複数の異なる文字コードで共通利用させるための前記複数種類の字形データと、前記複数の文字コードの描画の際に使用する字形データの種類を示すデータとを格納した電子文書を生成する生成ステップと、を有し、
前記電子文書に格納される、前記複数の文字コードの描画の際に使用する字形データの種類を示すデータは、前記文書画像内の各文字画像に対して前記選択ステップで選択された字形データの種類に基づいて記述されることを特徴とする画像処理方法。
【請求項１６】
コンピュータを、
文書画像内の複数の文字画像に対して文字認識処理を行うことにより、それぞれの文字画像に対応する文字コードを得る文字認識手段、
前記文書画像内の各文字画像についての文字の幅と文字行の高さとに基づいて、複数種類の字形データの中から、使用する字形データの種類を選択する選択手段、
前記文書画像と、前記文字認識手段で得た複数の文字コードと、前記複数の文字コードに対応する文字を描画する際に複数の異なる文字コードで共通利用させるための前記複数種類の字形データと、前記複数の文字コードの描画の際に使用する字形データの種類を示すデータを格納した電子文書とを生成する生成手段、
として機能させるための、コンピュータプログラムであって、
前記電子文書に格納される、前記複数の文字コードの描画の際に使用する字形データの種類を示すデータは、前記文書画像内の各文字画像に対して前記選択手段で選択された字形データの種類に基づいて記述されることを特徴とするコンピュータプログラム。
【請求項１７】
コンピュータを、
文書画像内の複数の文字画像に対して文字認識処理を行うことにより、それぞれの文字画像に対応する文字コードを得る文字認識手段、
前記文書画像内の各文字画像についての文字送り幅と文字行の高さとに基づいて、複数種類の字形データの中から、使用する字形データの種類を選択する選択手段、
前記文書画像と、前記文字認識手段で得た複数の文字コードと、前記複数の文字コードに対応する文字を描画する際に複数の異なる文字コードで共通利用させるための前記複数種類の字形データと、前記複数の文字コードの描画の際に使用する字形データの種類を示すデータとを格納した電子文書を生成する生成手段、
として機能させるための、コンピュータプログラムであって、
前記電子文書に格納される、前記複数の文字コードの描画の際に使用する字形データの種類を示すデータは、前記文書画像内の各文字画像に対して前記選択手段で選択された字形データの種類に基づいて記述されることを特徴とするコンピュータプログラム。
【請求項１８】
請求項１６または１７のいずれかに記載のコンピュータプログラムを記憶した、コンピュータ読取可能な記憶媒体。
【発明の詳細な説明】
【技術分野】

【０００１】
本発明は、紙文書のスキャン画像を電子的に検索可能なデータへ変換する技術に関する。
【背景技術】
【０００２】
近年、スキャナおよびハードディスク等大容量記憶装置の普及により、これまで紙で保存されていた文書をスキャンし、電子文書として保存されるようになっている。その際、紙文書をスキャンして得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、文書に記載されている文字情報を読みとり、画像にその文字情報を関連付けて保存しておくことも行われている。ユーザは、このようにして文字情報が関連付けられた電子文書を、検索キーワードを用いて検索できる。このように、大量の保存文書群の中から所望の文書を高速に検索するためには、スキャン画像に対してもキーワード検索できるようにすることが重要である。
【０００３】
例えば、特許文献１では、このような文字情報が関連づけられた電子文書に対して、ユーザが検索キーワードを用いて検索した際、その文書画像上で当該検索キーワードが記載されている部分をユーザが識別できるように強調表示することが記載されている。このように、検索キーワードに対応する文字部分が強調された状態で表示されるので、文書内に同じキーワードの記載箇所が複数ある場合でも、ページ画像を切り替えていくことにより、ユーザは効率よく記載部分を識別することができる。
【０００４】
また一方、文字認識処理した結果を透明テキスト（描画色として透明色が指定された文字コード）として画像ファイル中に埋め込み、ＰＤＦ（Ｐｏｒｔａｂｌｅ　Ｄｏｃｕｍｅｎｔ　Ｆｏｒｍａｔ）形式で保存する技術もある。このように作成されたＰＤＦファイルを表示させると、文書画像内の文字画像上に透明なテキストが描画されることになる。したがって、キーワード検索を行うと、透明テキストが検索されるが、ユーザにとって透明テキスト自体は見えていないので、あたかも画像が検索されているかのように見えることになる。このようにすれば、画像と文字の描画が可能なページ記述言語で記述されたフォーマットのファイルにもとづき、検索キーワードで検索可能な画像を描画することができる。
【特許文献１】特開２０００－３２２４１７
【発明の開示】
【発明が解決しようとする課題】
【０００５】
ＰＤＦやＳＶＧなどのページ記述言語を用いた電子文書における文字の描画には、各文字の字形情報、すなわちフォントデータが必要である。しかしながら、フォントデータは一般にサイズが大きいため、電子文書のサイズを小さくするためには、電子文書内にフォントデータを格納せずに、電子文書内には、フォントの種類の指定をおこなっておくことが一般に行われている。このようにすれば、アプリケーションで描画する際に、パソコンにインストールされているフォントを利用して描画することができる。
【０００６】
一方、電子文書内にフォントデータを格納しておくことが望まれる場合もある。例えば、文書作成アプリケーションで作成した電子文書を他のパソコンで開く場合、当該電子文書で使用されているフォントデータがそのパソコンにインストールされていなければ、その電子文書を正確に開くことはできない。言い換えると、指定のフォントデータをインストールしていないパソコンやアプリケーションで電子文書を再生する場合であっても、フォントデータ自体が電子文書内に格納されていれば、該電子文書を正確に再生することができる。
【０００７】
また、用途によっては、文字の描画に使用するフォントデータを電子文書内に格納して

おくのを必須条件にした方がいい場合もある。例えば、長期保存対象のファイルなどは、長期間経過後、ＯＳが変更されるなどして、デフォルトでインストールされているフォントが変更になることも考えられるので、フォントデータを格納する形式を必須にしておくのがよいと考えられる。

【０００８】

また、フォーマットの形式によっては、フォントデータを電子文書内に格納しておくことが必須条件になっているフォーマットも存在する。例えば、ＸＰＳ（ＸＭＬ Ｐａｐｅｒ Ｓｐｅｃｉｆｉｃａｔｉｏｎ）のフォーマットでは、テキストデータを保存する場合、フォントデータも一緒に格納しておく必要がある。

【０００９】

しかしながら、電子文書内にフォントデータを格納すると、電子文書のサイズ自体が増加してしまう。ファイルサイズが増加すると、電子文書をネットワークで送信する際の時間が多くかかってしまったり、保存する場合の記憶容量が多く必要になったりしてしまうという問題がある。

【００１０】

このように電子文書内に格納されているフォントデータを用いて描画するファイル形式の電子文書において、ファイルサイズの増加を防ぐことが望まれることになる。特に、スキャン画像と、文字認識処理した結果のテキストデータと、テキスト描画用のフォントデータとを一緒に電子文書内に格納する場合に、ファイルサイズの増加を防ぐことが望まれる。フォーマットの制約やシステム上の制約などにより電子文書内にフォントデータを格納しなければならないようなとき、ファイルサイズの増加は問題になりやすい。

【００１１】

また、文書画像に文字認識結果を透明テキストとして埋め込む場合、検索時に検索されたテキストの位置と文字画像の位置とが合うように、透明テキストの描画位置と文書画像内の対応する文字画像の位置とを正確に合わせるのが望ましい。正確に位置を合わせようとすると、透明テキストの描画位置（文字の位置座標や文字幅や文字送り幅など）を文字ごとに細かく指定することが考えられる。しかしながら、全ての文字について、各文字の位置を別個に記述すると、特に文字数が多い場合などは、生成する電子文書のファイルサイズが大きくなりやすい。

【００１２】

本発明では、透明テキストの描画位置と、文書画像内の対応する文字画像の位置とが合うようにしつつ、ファイルサイズを抑えた電子文書を生成することを目的とする。

【課題を解決するための手段】

【００１３】

上記課題を解決するために、本発明の画像処理装置は、文書画像内の複数の文字画像に対して文字認識処理を行うことにより、それぞれの文字画像に対応する文字コードを得る文字認識手段と、前記文書画像内の各文字画像についての文字の幅と文字行の高さとに基づいて、複数種類の字形データの中から、使用する字形データの種類を選択する選択手段と、前記文書画像と、前記文字認識手段で得た複数の文字コードと、前記複数の文字コードに対応する文字を描画する際に複数の異なる文字コードで共通利用させるための前記複数種類の字形データと、前記複数の文字コードの描画の際に使用する字形データの種類を示すデータとを格納した電子文書を生成する生成手段と、を有し、前記電子文書に格納される、前記複数の文字コードの描画の際に使用する字形データの種類を示すデータは、前記文書画像内の各文字画像に対して前記選択手段で選択された字形データの種類に基づいて記述されることを特徴とする。

または、上記課題を解決するために、本発明の画像処理装置は、文書画像内の複数の文字画像に対して文字認識処理を行うことにより、それぞれの文字画像に対応する文字コードを得る文字認識手段と、前記文書画像内の各文字画像についての文字送り幅と文字行の高さとに基づいて、複数種類の字形データの中から、使用する字形データの種類を選択する選択手段と、前記文書画像と、前記文字認識手段で得た複数の文字コードと、前記複数

の文字コードに対応する文字を描画する際に複数の異なる文字コードで共通利用させるための前記複数種類の字形データと、前記複数の文字コードの描画の際に使用する字形データの種類を示すデータとを格納した電子文書を生成する生成手段と、を有し、前記電子文書に格納される、前記複数の文字コードの描画の際に使用する字形データの種類を示すデータは、前記文書画像内の各文字画像に対して前記選択手段で選択された字形データの種類に基づいて記述されることを特徴とする。

【発明の効果】

【００１４】

本発明によれば、描画するグリフを選択しておくことで、透明テキストの描画位置と文書画像内の各文字画像の位置とが合うようにできる。またグリフは、文字幅の異なる単純な字形（例えば矩形）をいくつか用意しておき、様々な字種を描画する際、共通して利用するようにしている。よって、記述するデータの量が少なくて済む。ゆえに、フォントデータを電子文書内に保存しなければならないような場合であっても、電子文書のファイルサイズを小さく抑えることができる。

【発明を実施するための最良の形態】

【００１５】

＜実施形態１＞

図１は画像処理装置の構成を示すブロック図の一例である。

【００１６】

画像処理装置１００は本実施形態を実現するための装置であり、文書画像データを検索可能な電子文書に変換する。画像処理装置１００は、スキャナ１０１、中央処理ユニット（ＣＰＵ）１０２、メモリ１０３、ハードディスク１０４、ネットワークインタフェース１０５、ユーザインタフェース（ＵＩ）１０６で構成される。スキャナ１０１は紙文書の紙面情報を読み取り、文書画像データに変換を行う。ＣＰＵ１０２は、画像データを解析して検索可能な電子文書へ変換するためのコンピュータプログラムなどを実行する処理部である。メモリ１０３は、該プログラムや処理中のデータを保持したり、ＣＰＵのワークスペースとして使用したりするための記憶媒体である。ハードディスク１０４は、該コンピュータプログラムや電子文書などのデータを格納するための大容量記憶媒体である。ネットワークインタフェース１０５は、ネットワーク１２０と接続するためのインタフェースであり、スキャン画像や前記変換された検索可能な電子文書などのデータを外部装置へ送信したり、外部装置からデータを受信したりするために使用される。ユーザインタフェース１０６は、ユーザからの指示を受け取るためのインタフェースであり、入力キーやタッチパネルなどの入力デバイスと、液晶などの表示デバイスから構成される。なお、本発明の装置の構成は、これに限るものではない。

【００１７】

画像処理装置１１０は、画像処理装置１００で作成された電子文書の検索や閲覧をおこなうことができる。ＣＰＵ１１１は、電子文書を検索したり閲覧したりするための処理をおこなうためのコンピュータプログラムを実行する。メモリ１１２は、該プログラムを実行するためのワークスペースとして使用したり、データを一時保存したりするための記憶媒体である。ハードディスク１１３は、コンピュータプログラムや電子文書などのデータを格納するための大容量記憶媒体である。ネットワークインタフェース１１４は、外部装置から電子文書などのデータを受信したり、外部装置へデータを送信したりするためのインタフェースである。ユーザインタフェース１１５は、ユーザからの指示を受け取るためのインタフェースであり、入力キーやタッチパネルなどの入力デバイスと、液晶などの表示デバイスから構成される。

【００１８】

次に、本実施形態１における処理を図２および図３のフローチャートを用いて説明する。

【００１９】

図２は、画像処理装置１００が、紙文書をスキャンするなどして取得した画像データか

ら検索可能な電子文書を生成し、画像処理装置１１０へ当該電子文書を送信する処理の例を示すフローチャートである。

【００２０】

ステップＳ２０１では、ユーザからの指示操作にしたがって、生成される電子文書の送信先と送信方法を決定する。ユーザからの指示はユーザインタフェース１０６を介して行われる。また、送信方法は、例えば、電子メール、ＦＴＰを用いたファイル転送、などの選択肢から選択される。

【００２１】

ユーザが紙文書をセットしてスタートキーを押下すると、ステップＳ２０２では、スキャナ１０１を用いて当該セットされた紙文書をスキャンして文書画像データを生成してメモリに保存する。なお、オートドキュメントフィーダなどを用いて、複数ページで構成される文書が入力された場合は、１ページ毎に１つのページ画像データへと変換され、入力順にメモリ１０３に保存されるものとする。

【００２２】

図７にページ画像の例を示す。図７中のページ画像７０１には、「あ１Ａ」という文字列７０２と、「かＢｃ」という文字列７０３、および写真７０４が存在する。なお、説明のために、写真７０４を黒の矩形で簡略的に示しているが、実際には自然画である。また、図７の例では、文字列７０２，７０３と、写真７０４の例しか示していないが、その他に、図形等の領域があっても構わない。

【００２３】

なお、ページ画像データの形式は、例えば、紙文書がカラーであれば、ＲＧＢ各々８ｂｉｔで階調を表現するカラー画像で扱い、紙文書が白黒であれば８ｂｉｔで輝度を表現するグレー画像もしくは１ｂｉｔで白黒を表現する二値画像で扱うものとする。

【００２４】

ステップＳ２０３では、メモリ１０３に保存された未処理のページ画像データを、処理対象画像として選択する。なお、複数ページの画像がある場合は、入力順にしたがって１ページの画像を処理対象として選択する。

【００２５】

ステップＳ２０４では、処理対象の画像を解析して、テキスト領域、図領域、写真領域、表領域などといった性質の異なる領域ごとに領域識別する領域解析処理を行い、識別された各領域に関する領域データを生成してメモリ１０３に保存する。ここで領域データには、各領域の外接矩形の左上位置座標（ｘ，ｙ座標値）と、該外接矩形のサイズ（幅と高さ）を表わす画素数の値と、当該判別された領域の種別とが含まれるものとする。なお、前記領域解析処理には、公知の技術（領域識別処理、領域判別処理、領域抽出処理などとも言う）を用いるものとする。例えば、特開平６－６８３０１号公報に開示される技術を用いれば、二値化した文書画像データから、似たような大きさの黒画素塊が縦または横に連なる範囲をテキスト領域として抽出することができる。

【００２６】

図７のページ画像７０１に対して領域解析処理を行った結果、図８のように、テキスト領域８０１、写真領域８０２とが識別される。図９は、その領域解析処理で得られた領域データの例である。

【００２７】

ステップＳ２０５では、領域解析処理で識別された各テキスト領域内の文字画像に対して文字認識処理をおこなうことにより、各テキスト領域についての文字コード列のデータを得て、メモリ１０３に保存する。ここで、文字コード列のデータには、テキスト領域内に含まれる各文字画像に対する認識結果である文字コード情報と、各文字画像の外接矩形の情報（外接矩形の左上座標と幅と高さの情報、ならびに該文字が含まれる行の高さの情報）とが含まれるものとする。

【００２８】

文字認識処理の一例を簡単に説明する。なお、文字画像を文字認識する処理は、公知の

技術を利用することが可能である。

【００２９】

まず、文書画像が二値画像でない場合はテキスト領域内を二値化するなどして、テキスト領域内の二値画像を得る。当該二値化された各テキスト領域内について、縦横のライン毎の黒画素数を計数してヒストグラムを作成する。縦横のヒストグラムに基づいて、周期的なヒストグラムになっている方向を行方向とし、ヒストグラムの黒画素数が所定の閾値以上になる部分を文字行を構成する部分として、短冊状の行画像を得る。次に、各行画像に対して、行方向と垂直な方向でヒストグラムをとり、ヒストグラムの結果に基づいて１文字ずつの画像を切り出す。この切り出された範囲が１文字の外接矩形情報となる。なお、ここでは、黒画素数を計数したヒストグラムを用いて判別を行ったが、各ラインに黒画素があるかないかを示す射影を用いて文字領域の判別を行うようにしてもよい。

【００３０】

次に、各文字画像の外接矩形内の画像から、エッジ成分などを取り出して特徴ベクトルを得て、あらかじめ登録された文字認識用辞書内の特徴ベクトルと比較し、類似度を求める。そして、最も類似度の高い字種（文字種）のコードを、当該矩形内の文字画像に対する文字コードとする。このようにして、テキスト領域内に存在する全ての文字の外接矩形に対して、文字コードを割り当てたデータが得られる。そして、各テキスト領域から得た文字コード群が文字コード列となる。

【００３１】

また、英文の文字領域に対しては、文字間に単語間スペースが存在するか否かの判定も行うこととする。例えば、文字間の距離が広いかどうかや、文字画像の文字認識結果の文字列と単語辞書とのマッチングを行って単語の切れ目であるかどうかなどを判別することにより、単語間スペースが存在するかどうか判定することができる。単語間スペースが存在すると判定した場合は、当該スペースの文字コードを文字コード列に挿入することになる。

【００３２】

図１０及び図１１は、図８のテキスト領域８０１に対して文字認識処理を行った例を示す。図１０中のテキスト領域１０００から文字行１００１と１００２が先ず切り出される。そして、文字行１００１内から１０１１，１０１２，１０１３の３文字が切り出されて、それぞれ認識処理が行われる。その結果、各文字に対応する文字コードが得られて、図１１の１１０１に示したような文字コード列データが生成される。同様に、文字行１００２内から切り出された１０２１，１０２２，１０２３の３文字にも文字認識処理が実行され、図１１の１１０２に示したような文字コード列データが生成される。

【００３３】

なお、上記説明は一例であって、他の公知の文字認識技術を利用した処理方法を用いて、文字コード列を取得してもよい。

【００３４】

ステップＳ２０６では、当該処理対象となっているページ画像データと領域データと文字コード列データとを関連付けて、メモリ１０３もしくはハードディスク１０４に一時保存する。

【００３５】

ステップＳ２０７では、未処理の画像データがあるかどうかを判定し、あればステップＳ２０３に戻り、次のページ画像データの処理を行う。なければ、ステップＳ２０８に進む。

【００３６】

ステップＳ２０８では、メモリ１０３あるいはハードディスク１０４に保存された全ページ分のデータをページ順に合成して、複数ページからなる検索可能な電子文書を生成する。

【００３７】

このステップＳ２０８で生成される電子文書のデータは、各ページ画像をディスプレイ

等に電子的に表示あるいはプリンタにより印刷する為の描画情報と、検索キーワードで検索できるようにするための内容情報の両方を保持可能なデータである。そのような条件を満たすデータフォーマットとしては、ＰＤＦ、ＳＶＧなどがあるが、本実施形態では、このとき生成される電子文書のフォーマットとして、フォントデータを埋め込むことが指定されていたとする。なお、フォントデータを埋め込むことが必須条件になっているフォーマット形式としては、例えば、ＸＰＳなどがある。以下では、ＸＭＬ表現を用いたページ記述フォーマットの仕様を仮定しながら説明するが、本発明はこのフォーマットに限るものではない。

【００３８】

図６は、２ページ分のページ画像で構成される文書が入力された場合に、本説明で用いるページ記述フォーマットの仕様に基づいて生成された電子文書のページ記述例である。なお、ここでは、ページ記述フォーマットの例として、図６に示したように、１つのファイル内にまとめて記述するものとするが、これに限るものではない。例えば、フォントデータの部分を別ファイルにして、本体のファイルからフォントデータファイルを参照するようにし、それらをＺＩＰ圧縮等で１つの電子文書にまとめるようなフォーマット（例えば、ＸＰＳ）でもよい。

【００３９】

以下に、ステップＳ２０８にて行われる電子文書データ生成処理の例を、図４のフローチャートを用いて説明する。

【００４０】

ステップＳ４０１では、電子文書の開始タグの記述を行う。本説明のページデータ記述フォーマット仕様では、＜Ｄｏｃｕｍｅｎｔ＞という要素が電子文書の開始タグを表すものとする。なお、その＜Ｄｏｃｕｍｅｎｔ＞の終了を示す＜／Ｄｏｃｕｍｅｎｔ＞までに挟まれた範囲のＸＭＬ記述が、当該文書に含まれる各ページに関する記述データとなる。図６の例では６０１が電子文書の開始タグ、６１２が終了タグを表す。

【００４１】

ステップＳ４０２では、未記述のページのうち、先頭ページに関するデータを特定して処理対象とする。

【００４２】

ステップＳ４０３では、処理対象ページデータの開始を表わすタグを生成して記述する。本例では＜Ｐａｇｅ＞という要素タグがページデータの開始を表わし、その終了タグとなる＜／Ｐａｇｅ＞までに挟まれた範囲のＸＭＬ記述が、当該ページ内の描画データおよび内容データとなる。また、＜Ｐａｇｅ＞タグには、当該ページの画素幅と高さを示す属性ＷｉｄｔｈとＨｅｉｇｈｔ、ならびに解像度を示す属性Ｄｐｉを用いてページの物理的な大きさが記述され、また、ページ番号を示す属性Ｎｕｍｂｅｒを用いてページ番号が記述される。

【００４３】

図６の記述例では、＜Ｐａｇｅ＞要素の開始タグ６０２に、当該ページの幅Ｗｉｄｔｈが“１６８０”、高さＨｅｉｇｈｔが“２３７６”、解像度Ｄｐｉが“２００”であり、ページ番号Ｎｕｍｂｅｒが“１”であることが記述されている。また、当該１ページ目のデータは、終了タグ６０６までの間（６０３～６０６）に記述されている。

【００４４】

ステップＳ４０４では、ページを構成するデータのうち、画像の描画データを表わすタグを生成して記述する。

【００４５】

本説明のページデータ記述フォーマット仕様では、１つの＜Ｉｍａｇｅ＞要素が１つの画像の描画データを表わすものとする。また、画像データの内容を属性Ｄａｔａ内に記述し、その画像がページ内に描画される位置を属性Ｘ，Ｙ，Ｗｉｄｔｈ，Ｈｅｉｇｈｔの座標情報で記述するものとする。ページ内に画像が複数ある場合は、各画像データを登場順に上へ重ね書きしていくことを意味する。なお、属性Ｄａｔａ内には圧縮された画像デー

タ形式で記述されるものとし、ここでは、圧縮方式として、カラー・グレーの場合はＪＰＥＧ圧縮、二値の場合はＭＭＲ圧縮したコード列を用いるものとする。

【００４６】

図６の記述例６０３では、文書の１ページ目のスキャン画像が全面にわたって描画されるようにしている。図６のタグ６０３では、画像の位置とサイズを「Ｘ＝“０”、Ｙ＝“０”、Ｗｉｄｔｈ＝“１６８０”、Ｈｅｉｇｈｔ＝“２３７６”」として記述している。また、画像をＪＰＥＧ圧縮して生成されたコード列の文字列を、属性Ｄａｔａの値として記述している（なお、図６では、図を単純に示すため、Ｄａｔａ属性の文字列を一部省略して示している）。このようにして、＜Ｉｍａｇｅ＞要素６０３が記述されている。なお、スキャン画像をＪＰＥＧ圧縮して保存する前に、必要に応じて、解像度を変更して保存するようにしてもよい（例えば６００ｄｐｉでスキャンした画像を３００ｄｐｉに変更して保存してもよい）。

【００４７】

ステップＳ４０５では、ページを構成するデータのうち、文字の描画データを表わす記述を生成する。

【００４８】

本説明のページ記述フォーマット仕様では、１つの＜Ｔｅｘｔ＞要素が１行分の文字の描画データを表わしている。また、＜Ｔｅｘｔ＞要素内に記述される属性データは、Ｄｉｒｅｃｔｉｏｎ、Ｘ，Ｙ、Ｆｏｎｔ、Ｓｉｚｅ、Ｃｏｌｏｒ、Ｓｔｒｉｎｇ、ＣＧｌｙｐｈＩｄなどがある。ここで、属性Ｄｉｒｅｃｔｉｏｎは、文字列が縦書きか横書きかを示し、Ｄｉｒｅｃｔｉｏｎ属性が記述されていない場合はデフォルトの方向（例えば、左から右に向かう水平方向）が使用される。また、属性Ｘ，Ｙは、文字の開始位置の座標を指定する。属性Ｆｏｎｔは、文字コードを描画するためのフォントデータのＩＤを指定する。属性Ｓｉｚｅは、フォントサイズを指定する。また、属性Ｃｏｌｏｒは、描画時の文字色を、Ｒ成分値，Ｇ成分値，Ｂ成分値、透過度を表すアルファチャネル値の４値組で指定する。また、属性Ｓｔｒｉｎｇは、文字列の内容（文字コード列）を指定する。また、属性ＣＧｌｙｐｈＩｄは、Ｓｔｒｉｎｇ内の各文字が描画の際に使用する字形データすなわちグリフのＩＤを指定する。なお、Ｄｉｒｅｃｔｉｏｎが指定されていない場合は、デフォルトで横書きとする。

【００４９】

＜Ｔｅｘｔ＞要素を構成する文字コード列は、図２のステップＳ２０５で生成された文字コード列のデータを、文字行毎、すなわち縦または横に連なる文字の集合に更に分割したものが使用される。

【００５０】

図６の記述例では、２つの＜Ｔｅｘｔ＞６０４および６０５は、１ページ目の文字描画記述に関するものであり、図１１の文字コード列データ１１０１および１１０２それぞれに対応する記述である。例えば、図１１の１１０１の３文字の横書き文字列「あ１Ａ」に対応する＜Ｔｅｘｔ＞要素６０４では、下記のような属性が指定されている。

【００５１】

属性Ｘ，Ｙには、３文字分の外接矩形の左上座標としてＸ＝“２３６”、Ｙ＝“２７２”が指定されている。

【００５２】

フォントの種類を示す属性Ｆｏｎｔには、“Ｆｏｎｔ０１”が指定されている。また、フォントサイズを示す属性Ｓｉｚｅには、当該文字行内の文字の高さから類推して“９７”ピクセルが指定されている。描画時の文字色を示す属性Ｃｏｌｏｒには、Ｒ成分値＝Ｇ成分値＝Ｂ成分値＝０とアルファチャネル＝２５５とが指定されている（つまり、透明色が指定されている）。

【００５３】

また、文字列の内容（各文字に対応する文字コードの列）を示す属性Ｓｔｒｉｎｇには、“０ｘ２４２２，０ｘ２３３２，０ｘ２３４１”とが指定されている。

【００５４】
属性ＣＧｌｙｐｈＩｄには、各文字の字形データとして用いるグリフのＩＤが指定される。ここでは、ステップＳ２０５で得た各文字の幅情報に基づきグリフのＩＤが指定されるものとする。なお、本実施形態では、スキャン画像上に透明色の文字の字形を描画するようにしているので、ユーザの視覚には見えていない。そこで、本実施形態では、字形データとして、文字画像の形状そのままの字形を用いるのではなく、単純な形状（例えば矩形）の字形データをいくつか（例えば８種類）用意しておき、その中から使用する字形データ（グリフのＩＤ）を選択して用いることとする。つまり、グリフとして、矩形幅のサイズが異なるグリフ（縦横比の異なるグリフ）を複数用意しておき、その中から文字毎に適したグリフを選択するように制御する。したがって、本実施形態では、文字が含まれている行の高さと各文字の文字幅との比に基づいて、グリフのＩＤが選択されるものとする。図１５は、画像処理装置１００が、各文字画像に対するグリフのＩＤを選択するためのフローチャートの例である。ステップＳ１５０１～Ｓ１５０７では、（文字幅／行高さ）を、（７／８）、（６／８）、（５／８）、（４／８）、（３／８）、（２／８）、（１／８）と比較する。その比較結果に応じて、ステップＳ１５０８～Ｓ１５１５のいずれかに進んで、グリフのＩＤ（０～７）を選択する。すなわち、（文字幅／行高さ）＞（７／８）であると判断された場合は、グリフＩＤ＝０と選択される。（７／８）＞＝（文字幅／行高さ）＞（６／８）であると判断された場合は、グリフＩＤ＝１と選択される。（６／８）＞＝（文字幅／行高さ）＞（５／８）であると判断された場合は、グリフＩＤ＝２と選択される。（５／８）＞＝（文字幅／行高さ）＞（４／８）であると判断された場合は、グリフＩＤ＝３と選択される。（４／８）＞＝（文字幅／行高さ）＞（３／８）であると判断された場合は、グリフＩＤ＝４と選択される。（３／８）＞＝（文字幅／行高さ）＞（２／８）であると判断された場合は、グリフＩＤ＝５と選択される。（２／８）＞＝（文字幅／行高さ）＞（１／８）であると判断された場合は、グリフＩＤ＝６と選択される。（文字幅／行高さ）＜＝（１／８）であると判断された場合は、グリフＩＤ＝７と選択される。なお、この例では、グリフＩＤの番号が小さい方が、より矩形幅が広いグリフになっている。例えば、図１１の１１０１の文字列では、（文字幅／行高さ）がそれぞれ「０．８２」、「０．１４」、「０．５７」であるので、図１５の選択処理により、グリフのＩＤは、“１，６，３”と指定されることになる。なお、英文の単語間スペースに関しても同様に、スペースの幅を文字幅として扱うことにより、当該スペースが含まれている文字行の高さと当該スペースの幅との比に基づいて、グリフＩＤが選択される。なお、グリフの形状の詳細については、後述する。

【００５５】
なお、上記の属性値は一例であって、同様な意味を持つ別の値で記述してもよい。例えば、フォントサイズの属性サイズは、ピクセル高さと画像解像度に基づき、画素数ではなくポイント数等の値で記述されてもよい。

【００５６】
ステップＳ４０６では、当該ページの終了を示す＜／Ｐａｇｅ＞を記述する。

【００５７】
ステップＳ４０７では、未記述のページが他に有るか否かを判定し、未記述のページがある場合は、次のページを処理対象のページ画像としてステップＳ４０３に戻る。一方、未記述のページがない場合は、ステップＳ４０８に進む。

【００５８】
図６の記述例では、２ページ目の画像に対してもステップＳ４０４～Ｓ４０６の処理が行われ、６０７～６１０の部分が記述されることになる。

【００５９】
ステップＳ４０８では、この電子文書で文字列の描画に使用される全グリフを含むフォントデータの内容を記述する。

【００６０】
本説明のページデータ記述フォーマット仕様では、＜Ｆｏｎｔ＞と＜／Ｆｏｎｔ＞に挟

まれる範囲に、フォントデータに含まれるグリフデータが＜Ｇｌｙｐｈ＞要素として記述される。＜Ｆｏｎｔ＞要素には、当該フォントの種類を示す属性ＩＤが含まれる。また、＜Ｇｌｙｐｈ＞要素には、グリフの種類を示す属性ＩＤと、そのＩＤに対応するグリフ（字形）を示す属性Ｐａｔｈとが含まれる。ここで、属性Ｐａｔｈは、左下を原点とする描画矩形単位内で、直線や曲線関数を用いてグリフを表現するように記述される。

【００６１】

図６の記述例では、＜Ｆｏｎｔ＞要素６１１において、Ｉｄ＝“Ｆｏｎｔ０１”のフォントが定義され、その中に、グリフＩｄ＝“０”～“７”のグリフが８種類定義されている。例えば、Ｉｄ＝“７”のグリフの字形を表わすＰａｔｈ属性“Ｍ０，０　Ｖ－１０２４　Ｈ１２８　Ｖ１０２４　ｆ”は、「原点（０，０）にＭＯＶＥ，上方向に１０２４単位縦線を描画、右方向に１２８単位横線描画、下方向に１０２４単位縦線描画、現在の点から開始点まで線を描画して囲まれた範囲を塗りつぶす」というグリフを記述している。すなわち、１０２４×１２８の矩形を塗りつぶした矩形のグリフを表現する記述となっている。そのほかのＩＤはそれぞれＩＤ＝“７”を横方向に段階的に整数倍した長方形のグリフであり、例えば、ＩＤ＝“０”は１０２４×１０２４を塗りつぶす正方形のグリフを表現する記述となっている。

【００６２】

なお、図６の＜Ｆｏｎｔ＞要素６１１の記述は一例であって、三角や丸、直線などの他の単純な字形を定義してもよいし、空白（スペース形状）を字形として定義してもよい。

【００６３】

ステップＳ４０９では、電子文書の終了を示す＜／Ｄｏｃｕｍｅｎｔ＞を記述し、電子文書の生成を終了する。生成された電子文書はファイルとして画像処理装置１００内のメモリ１０３あるいはハードディスク１０４に保存される。保存の際には公知のテキスト圧縮技術を用いて圧縮を施してもよい。

【００６４】

図２に戻ってステップＳ２０９では、ステップＳ２０８で生成された電子文書を、ステップＳ２０１で指定された送信先（例えば画像処理装置１１０）へ、指定された送信方法で送信する。データ転送処理自体は公知技術を用いるものとして説明は省略する。

【００６５】

送信先の装置１１０では、ネットワークインタフェース１１４を介して転送されてきた電子文書を受信し、ハードディスク１１３に蓄積する。データ受信処理は公知技術を用いるものとして説明は省略する。

【００６６】

なお、装置内で蓄積される電子文書をハードディスク内部で特定するための識別情報（ファイル名など）は任意のものでよい。例えば、受信時刻に関連する文字列を付与すればよい。その他にも、重複しない番号を選択して自動付与したり、電子文書生成時にユーザが指定するようにしても構わない。

【００６７】

次に、電子文書を検索・閲覧する処理の例を図３のフローチャートに従って説明する。ここでは、画像処理装置１１０で検索を行う例について述べるが、これに限るものではなく、画像処理装置１００で検索を行えるようにしても構わない。

【００６８】

ステップＳ３０１では、画像処理装置１１０内に蓄積された電子文書群から所望の電子文書の文字列を検索するために、ユーザは当該電子文書のテキストに含まれていると考えた検索キーワードをＵＩ１１５より入力する。ここで入力された文字列の長さをｋとする。

【００６９】

ステップＳ３０２では、画像処理装置１１０のハードディスク１１４内にある全ての電子文書ファイルに対し、未検索の電子文書ファイルがあるか否か判断する。未検索の電子文書ファイルがあれば、その中の１つの電子文書ファイルを特定し、その電子文書ファイ

ルが圧縮されている場合は展開して、ステップＳ３０３に進む。未検索の電子文書がなければＳ３１２に進み、全ての電子文書に対する検索が終了したことをユーザに報知する。

【００７０】

ステップＳ３０３では、Ｓ３０２で特定された電子文書内のテキストデータを対象にして検索を行うための準備を行う。ここでは、文書内のテキスト（文字コード）を１列に並べ、探索開始位置ｎを初期化、すなわちｎ＝０に設定する。

【００７１】

ステップＳ３０３の処理例を以下に説明する。まず、電子文書データをＸＭＬパーサでパースしていき、＜Ｔｅｘｔ＞要素が表われたら属性Ｓｔｒｉｎｇに記述されている文字コード列を取得する。そのＳｔｒｉｎｇ属性中に記載された文字コード列に基づいて、１文字ずつ、当該文字コードとその文字コード値が該電子文書データ中で記述されている位置との組を、文字コード配列テーブルに追加していく。ここで、文字コード値が記述されている位置とは、電子文書データ中で該文字コードが記述されているキャラクタ列の先頭が、該電子文書データの先頭から数えて何キャラクタ目であるかを示す値である。図６の電子文書から生成した文字コード配列テーブルの例を図１２に示す。例えば、図６の電子文書内の＜Ｔｅｘｔ＞要素６０４の属性Ｓｔｒｉｎｇに記述された３つの文字コード“０ｘ２４２２”、“０ｘ２３３２”、“０ｘ２３４１”は、それぞれこの電子文書の先頭から数えて１０９３キャラクタ目、１１００キャラクタ目、１１０７キャラクタ目の位置より記述されているものとする。同様に、６０５及び６０９に基づいて、残り６つの文字コードに対しても記述位置を求めて、図１２のような文字コード配列テーブルを生成する。なお、図１２では、このとき、文字列番号（Ｎｏ．）を０から順に付与している。

【００７２】

ステップＳ３０４では、文字コード配列テーブルに対して、探索開始位置ｎを起点として、検索キーワードの文字コード列と一致するか否か判断する。一致する部分を検出した場合、そのときの変数ｎを一致文字列の先頭位置としてステップＳ３０５に進む。

【００７３】

一方、ステップＳ３０４で一致しないと判断した場合は、ステップＳ３０９に進み、当該文字コード配列テーブルの全ての文字を探索したか判断する。文字コード配列テーブルに格納されている文字コード列全ての探索が終了したと判断した場合はステップＳ３１１に進み、現在探索対象となっている電子文書の検索が終了したことを報知する。一方、全ての探索が終了していないと判断した場合は、ステップＳ３１０に進んで、変数ｎを１インクリメントして、ステップＳ３０４に戻り、次の探索開始位置ｎで検索キーワードと一致するか判断する。なお、ステップＳ３０９では、文字コード配列テーブルに格納されている文字コードの総数をＮとした場合、ｎ＜（Ｎ－ｋ）ならば全ての探索が終了していないと判断し、ｎ＞＝（Ｎ－ｋ）ならば探索終了と判断すればよい。

【００７４】

例えば、図１２の文字コード配列テーブルの例に対し、検索キーワード「かＢ」の文字コード列“０ｘ２４２ｂ”，“０ｘ２３４２”を先頭から走査して一致する部分を探した場合、Ｓ３０４、Ｓ３０９、Ｓ３１０の処理が繰り返されて、最初の一致文字列の文字列番号としてｎ＝３が抽出される。

【００７５】

ステップＳ３０５では、文字列番号ｎに相当する文字列データが、電子文書のどのページに属しているかを特定する。

【００７６】

例えば、電子文書データをパースする際に、＜Ｔｅｘｔ＞要素がどの＜Ｐａｇｅ＞要素に記述されているかを判別すれば、Ｎｕｍｂｅｒ属性によってページ番号が識別できる。したがって、ステップＳ３０５で特定した位置ｎに対応する文字列の記述位置を図１２から求め、当該記述位置がどの＜Ｐａｇｅ＞要素の間にあるかによって、当該文字列が属するページが特定できる。なお、ステップＳ３０３で電子文書データをパースする際に、各＜Ｔｅｘｔ＞要素がどの＜Ｐａｇｅ＞要素に記述されているかを判別して、図１２の文字

コード配列テーブルに予め格納しておけば、文字列番号に基づいてページ番号が容易に特定できる。なお、ステップＳ３０４の一致文字列の検出方法や、ステップＳ３０５のページ番号の特定方法は、上述した例に限るものではない。

【００７７】

ステップＳ３０６では、ステップＳ３０５で決定されたページの描画記述に従って、当該ページの描画をおこなってＵＩ１１５に表示する。このとき、文字列番号（Ｎｏ．）がｎ～ｎ＋ｋ－１の範囲にある文字を描画する際には、その文字に対応する個所をユーザが識別しやすいように、該文字に強調効果を付けて描画する。この検索キーワードに一致する部分に強調効果を付けた描画の詳細については下記で説明する。

【００７８】

ステップＳ３０６で実施されるページの描画処理を、図５のフローチャートに従って説明する。

【００７９】

ステップＳ５０１では、特定されたページ番号に対応する＜Ｐａｇｅ＞要素のＷｉｄｔｈ，Ｈｅｉｇｈｔ属性の値から、描画結果となるページ画像のサイズを決定する。

【００８０】

ステップＳ５０２では、ページ画像の画素情報が格納できる分のメモリを確保する。

【００８１】

ステップＳ５０３では、当該＜Ｐａｇｅ＞要素の子要素の中で未処理の要素を１つ抽出し、当該未処理要素の種類を判別する。当該未処理要素が＜Ｉｍａｇｅ＞であると判別した場合は、ステップＳ５０４に進み、当該未処理要素が＜Ｔｅｘｔ＞であると判別した場合は、ステップＳ５０５に進む。当該＜Ｐａｇｅ＞要素の全ての子要素が既に処理されていたら、ステップＳ５１７へ進む。

【００８２】

ステップＳ５０４では、まず、＜Ｉｍａｇｅ＞要素のＤａｔａ属性値として記述されている圧縮画像を展開する。更に、Ｘ，Ｙ，Ｗｉｄｔｈ，Ｈｅｉｇｈｔの各属性により表されるページ画像内の描画矩形領域いっぱいに収まるように、当該展開されたイメージを変倍して、ステップＳ５０２で確保したページ画像メモリの該当領域へと上書きする。その後、ステップＳ５０３に戻る。

【００８３】

ステップＳ５０５では、処理対象の＜Ｔｅｘｔ＞要素に記述された各属性から、文字開始位置（Ｘ，Ｙ）、文字フォントＩＤ（Ｆ）、文字サイズ（Ｓ）、文字色（Ｃ）を取得する。また、当該＜Ｔｅｘｔ＞要素に記述された文字の数（Ｎ）も取得する。

【００８４】

ステップＳ５０６では、グリフ画像生成のためのメモリを確保する。ここでは１０２４×１０２４画素分の二値画像用メモリを確保するものとする。

【００８５】

ステップＳ５０７では、処理中の文字を示すカウンタｉを１に初期化する。

【００８６】

ステップＳ５０８では、ｉ＞Ｎであるか否かの判断を行い、ｉ≦Ｎの場合はステップＳ５０９に進み、ｉ＞Ｎの場合は当該＜Ｔｅｘｔ＞要素の処理は終了したと判断してステップＳ５０３に戻る。

【００８７】

ステップＳ５０９では、＜Ｔｅｘｔ＞要素の属性Ｓｔｒｉｎｇからｉ文字目の文字コード（Ｐ）と、属性ＣＧｌｙｐｈＩｄからｉ文字目のＧｌｙｐｈＩｄ（Ｑ）とを取得する。

【００８８】

ステップＳ５１０では、電子文書から、フォントＩｄが（Ｆ）である＜Ｆｏｎｔ＞要素記述を探し出し、更に、その＜Ｆｏｎｔ＞要素記述の子要素の中で、グリフＩｄが（Ｑ）である＜Ｇｌｙｐｈ＞要素からＰａｔｈ属性を取得する。

【００８９】

ステップＳ５１１では、ステップＳ５１０で取得したＰａｔｈ属性値にしたがって、ステップＳ５０６で確保したグリフ画像生成用メモリにおいてグリフの二値画像を生成する。なお、グリフの二値画像とは、例えば、描画が行われる部分を１、描画が行われない部分を０として表した画像である。なお、本実施例では、描画が行われる部分１は、後に、透明色で描画されることになる。

【００９０】

ステップＳ５１２では、グリフの二値画像を、文字サイズ属性の値（Ｓ）に則した大きさの矩形サイズになるよう変倍する。このとき、変倍後のグリフ二値画像の描画が行われる部分１の幅を、変倍グリフ幅Ｗｉとして取得する。

【００９１】

ステップＳ５１３では、ページ画像メモリ中の座標位置（Ｘ，Ｙ）を基準とした矩形領域に、ステップＳ５１２で変倍されたグリフの二値画像を描画する。ページ画像上に二値画像を重ねて描画したときの各画素の画素値を以下の式で定義する。なお、グリフを描画する前のページ画像の各画素値（ｒ，ｇ，ｂ）に対して、グリフを描画した後の画素値は（ｒ’，ｇ’，ｂ’）になるものとする。

グリフ二値画像の画素値が０に対応する画素：（ｒ’，ｇ’，ｂ’）＝（ｒ，ｇ，ｂ）

グリフ二値画像の画素値が１に対応する画素：（ｒ’，ｇ’，ｂ’）＝（Ｆ（ｒ，Ｃｒ），Ｆ（ｇ，Ｃｇ），Ｆ（ｂ，Ｃｂ））

ここで、Ｆ（ｒ，Ｃｒ）＝（ｒ×Ａ＋Ｃｒ×（２５５－Ａ））／２５５、Ｆ（ｇ，Ｃｇ）＝（ｇ×Ａ＋Ｃｇ×（２５５－Ａ））／２５５、Ｆ（ｂ，Ｃｂ）＝（ｂ×Ａ＋Ｃｂ×（２５５－Ａ））／２５５とする。また、Ａは文字色Ｃに対するアルファチャネル値、Ｃｒ，Ｃｇ，Ｃｂは文字色Ｃの各ＲＧＢ値とする。なお、アルファチャネル値として２５５が指定されている場合は、当該グリフ二値画像は透明であるので、グリフ二値画像の画素値が１に対応する画素についても、（ｒ’，ｇ’，ｂ’）＝（ｒ，ｇ，ｂ）となる。

【００９２】

ステップＳ５１４では、処理中のｉ文字目の文字が、文字列番号（Ｎｏ．）がｎ～ｎ＋ｋ－１の範囲にある文字であるか否かを、例えば、図１２の文字コード配列テーブルを用いて判定する。具体的には、範囲ｎ～ｎ＋ｋ－１内の各文字の記述開始位置が文字コード配列テーブルから分かるので、処理中の文字ｉの開始位置がそのいずれかに一致しているか否かに基づいて判定する。範囲ｎ～ｎ＋ｋ－１内の文字である場合はステップＳ５１５、それ以外の場合はステップＳ５１６に進む。

【００９３】

ステップＳ５１５では、処理中の文字が検索文字列として検出された範囲内にあることを示すための強調処理を行う。具体的には、当該文字列を描画した範囲に相当する、ページ画像メモリの位置（Ｘ，Ｙ）から始まる矩形領域内の各画素に対して、各画素値（ｒ，ｇ，ｂ）を以下の画素値（ｒ’，ｇ’，ｂ’）へと変更する。

（ｒ’，ｇ’，ｂ’）＝（Ｇ（ｒ），Ｇ（ｇ），Ｇ（ｂ））

（ここで、Ｇ（ｒ）＝２５５－ｒ，Ｇ（ｇ）＝２５５－ｇ，Ｇ（ｂ）＝２５５－ｂであるとする。）

なお、色の反転を行う上記強調処理は一例であり、その他の強調処理でもよい。例えば、グリフ二値画像の画素値が０の画素に対応する画素はそのまま変更せず、グリフ二値画像の画素値が１の画素に対応する画素については、各画素値（ｒ，ｇ，ｂ）を上記（ｒ’，ｇ’，ｂ’）にそれぞれ変更するようにしてもよい。

【００９４】

ステップＳ５１６では、次の文字の描画開始位置Ｘを決定し、ｎを１インクリメント（ｎ＝ｎ＋１）して、ステップＳ５０８に戻る。次の文字の描画開始位置Ｘは、現在の文字の描画開始位置に、変形グリフ幅Ｗｉと文字間の距離とを加算したもので計算される。本実施形態では、データ容量が少なくて済むように、文字送り幅や文字間の距離などのデータを保存していないので、文字間の距離は変形グリフ幅の１０％であると仮定して計算を

行うこととする。したがって、ここでは、次の文字の描画開始位置は、Ｘ＝Ｘ＋１．１Ｗｉで計算することとしている。なお、文字間の距離は、この計算に限るものではない。例えば、文字間の距離として、文字サイズＳの１０％としてもよいし、予め決めておいた定数であってもよい。

【００９５】

ステップＳ５１７では、１ページ分の描画結果、すなわち＜Ｐａｇｅ＞要素内の＜Ｉｍａｇｅ＞および＜Ｔｅｘｔ＞要素記述を描画したページ画像メモリの内容を、ＵＩ１１５の表示バッファに転送して表示させる。

【００９６】

以下では、図６の電子文書の１ページ目の描画記述を例として、図５のフローチャートの処理を実行した場合を説明する。

【００９７】

ステップＳ５０１の処理により、図６の１ページ目の＜Ｐａｇｅ＞要素６０２の属性値Ｗｉｄｔｈ＝“１６８０”、Ｈｅｉｇｈｔ＝“２３７６”に基づいて、ページの画像サイズを１６８０×２３７６ピクセルと決定する。

【００９８】

ステップＳ５０２の処理により、例えば、ページ画像がＲＧＢ２４ｂｉｔカラーで表現される場合、１６８０×２３７６×３バイトのメモリが確保される。

【００９９】

ステップＳ５０４の処理により、図６の＜Ｉｍａｇｅ＞要素６０３のＤａｔａ属性値に記述された圧縮のコードが展開されて画像になり、ページ画像メモリの全域に上書きされる。なお、本例では画像データは元のページと同サイズの１６８０×２３７６のピクセルの大きさを持っているので変倍処理は施されない。

【０１００】

次に、ステップＳ５０５の処理により、図６の＜Ｔｅｘｔ＞要素６０４から、Ｘ＝“２３６”，Ｙ＝“２７２”，文字数Ｎ＝“３”，文字フォントＩＤ＝“Ｆｏｎｔ０１”，文字サイズ＝“９７”，文字色＝“０，０，０，２５５”が得られる。

【０１０１】

ステップＳ５０９の処理により、まず最初は、＜Ｔｅｘｔ＞要素６０４のＳｔｒｉｎｇ属性の１番目の文字コード＝０ｘ２４２２およびＧｌｙｐｈＩｄ＝“１”が得られる。

【０１０２】

ステップＳ５１１でグリフの二値画像を生成するにあたって、まず、得られた文字フォントＩＤ＝“Ｆｏｎｔ０１”に基づき、当該ＩＤを有するグリフのＰａｔｈデータをステップＳ５１０で取得する。ここでは、図６の例では、＜Ｆｏｎｔ＞要素６１１内にある、＜Ｇｌｙｐｈ＞要素のＩｄ＝“１”のＰａｔｈ属性を取得する。そして、ステップＳ５１１において、当該取得した＜Ｇｌｙｐｈ＞要素のＩｄ＝“１”のＰａｔｈ属性のデータに基づいてグリフ画像を生成する。具体的には、Ｐａｔｈ属性の記述に従って、１０２４×８９６ピクセルの矩形領域すべてを１で塗りつぶした画像となる。つまり、グリフ画像生成用メモリとして確保した１０２４×１０２４の領域のうち、縦１０２４ピクセル、左端から横に８９６ピクセルで構成される長方形を１で塗りつぶすことにより、グリフ画像を生成する。

【０１０３】

ステップＳ５１２では、文字サイズ＝“９７”に基づいて、グリフ画像生成用メモリ１０２４×１０２４が９７×９７ピクセルに変倍される。したがって、塗りつぶされる領域は、縦９７×横８５ピクセル（変倍グリフ幅Ｗｉ＝８５）となる。

【０１０４】

ステップＳ５１３では、ページ画像上の位置（Ｘ，Ｙ）＝（２３６，２７２）から始まる９７×９７ピクセルの矩形範囲は変倍されたグリフの文字画像による描画対象領域となる。図６の例では文字色＝“０，０，０，２５５”すなわちアルファチャネル値Ａ＝２５５であるため、グリフの二値画像中の対応する画素値が１であっても常に（ｒ’，ｇ’，

ｂ’）＝（ｒ，ｇ，ｂ）となる。つまり、ステップＳ５１３の処理前後でページ画像内の当該矩形領域内の画素値は変化しない。

【０１０５】

ステップＳ５１４では、図６の＜Ｔｅｘｔ＞要素６０４内の１番目の文字が、文字列番号の範囲ｎ～ｎ＋ｋ－１内に相当する文字か否かを文字コード配列テーブルに基づいて判定する。

【０１０６】

ここでは、たとえば図６の電子文書から図１２の文字コード配列テーブルが生成されており、図３のステップＳ３０４でキーワードと一致すると判断された文字列の範囲が３～４であったとする。このとき、図６の＜Ｔｅｘｔ＞要素６０４内の１番目の文字コードは、範囲３～４でないので、ステップＳ５１６に進む。＜Ｔｅｘｔ＞要素６０４内の１番目の文字コード記述の先頭キャラクタ位置は１０９３であり、文字コード配列テーブルの文字列番号３～４の範囲の文字の記述位置のいずれとも一致しないことから、＜Ｔｅｘｔ＞要素６０４の１番目の文字は範囲３～４内に相当する文字でないと判定できる。

【０１０７】

その後、図６の＜Ｔｅｘｔ＞要素６０５内の１番目が示す文字の処理を行う際には、ステップＳ５１４において、文字コード配列テーブルの範囲３～４の文字の開始位置と一致すると判断し、ステップＳ５１５での強調描画処理が実行される。

【０１０８】

この＜Ｔｅｘｔ＞要素６０５内の１番目の文字に対するグリフＩＤは“０”のため、ページ画像メモリの位置（２３６，４７２）から始まる９２×９２の領域が透明色で塗りつぶされている。そこで、ステップＳ５１５では、ページ画像メモリの位置（２３６，４７２）から始まる９２×９２の領域内の各画素値（ｒ，ｇ，ｂ）を、（Ｇ（ｒ），Ｇ（ｇ），Ｇ（ｂ））へと変更させる。

【０１０９】

また、例えば、図６の＜Ｔｅｘｔ＞要素６０４内の１番目の文字コード（描画開始位置は（２３６，２７２）である）を描画した場合、ステップＳ５１６では、次の文字の描画開始位置Ｘとして、２３６＋１．１×８５＝３３０が計算される。したがって、＜Ｔｅｘｔ＞要素６０４内の２番目の文字コードの描画開始位置は、（３３０，２７２）となる。

【０１１０】

以上のようにして全＜Ｔｅｘｔ＞を描画した後、ページ画像は図１３のようになる。ステップＳ３０４で一致すると判定された範囲の文字に対応する領域に関しては、各矩形内で輝度が反転された状態となり、残りの文字に対応する領域は、＜Ｉｍａｇｅ＞要素が描画した画像データのままとなる。

【０１１１】

このように、検索した文字列が強調表示されるので、ページ内のどこに検索キーワードが存在するかを、ユーザはステップＳ３０６で表示されたページの画像を見るだけで容易に判断することができる。また、文字幅に合わせたグリフを用いて透明色で描画するため、検索時に強調される文字と文書画像内の当該文字画像との位置が合いやすくなり、ユーザは判別しやすくなる。

【０１１２】

図１４は、別の方法で強調表示をおこなうように設定した場合、どのようにページ画像表示がなされるかの例を示している。図１４（ａ）のページ描画記述では、フォントデータのグリフとして、文字高さ１０２４の下部の位置に、高さが１２８で幅が１０２４～１２８の塗りつぶし矩形を描画する８種類のグリフを定義している。図４のステップＳ４０５で＜Ｔｅｘｔ＞要素の属性データを記述する際、対応する文字画像の下部に相当する位置に、各グリフに対応する高さの低い矩形の透明文字が描画される。このようなページ描画記述に対し、ステップＳ５１５での強調処理において、各グリフの矩形範囲が反転強調されるようにすれば、図１４（ｂ）のように強調表示されたページ画像が生成される。このように、ユーザにとっては、検索した部分が下線（アンダーライン）を引かれて強調さ

れているかのように見えることになり、ユーザは検索した文字列がページ内のどこに存在するかを容易に判断することができる。

【０１１３】

図３に戻って、ステップＳ３０７では、検索・閲覧処理を終了するか、あるいは更に別の検索箇所を対象に検索を継続するかどうかをユーザに選択させる。ユーザが終了を選択した場合は、図３の処理を終了し、継続を選択した場合はステップＳ３０８に進む。

【０１１４】

ステップＳ３０８では、ｎ＝ｎ＋ｋとし、ステップＳ３０４に戻って、次に検索キーワードと一致する部分を検索する。

【０１１５】

以上説明したように、本発明の実施形態１によれば、紙の文書が電子文書へと変換される際に、ページ画像上にページから抽出した文字が透明色で描画されるように記述される。この電子文書に対しては、検索キーワードに一致する箇所が強調表示されたページ表示を確認しながら検索を進めていくことが可能である。

【０１１６】

この電子文書は、いくつかの文字幅の異なる単純な字形（例えば矩形）からなるフォントデータを内部で持ち、文書内の様々な字種の透明文字を描画する際に、各字種の幅に適合する単純な字形を選択して描画するように記述している。つまり、多数の字種（例えば数百種類の字種）に対して、数種の字形（例えば８種の幅の異なる字形）を共通して利用するようにしている。また、透明テキストの描画位置を全ての文字に対して文字ごとに（文字の位置座標などを用いて）細かく記述しなくても、透明テキストの描画位置と文書画像内の各文字画像の位置とがほぼ合うようになる。したがって、電子文書内で使用されるフォントデータを当該電子文書内に保存しなければならないような場合であっても、電子文書のファイルサイズ（データ容量）を小さく抑えることができる。

【０１１７】

＜実施形態２＞

実施形態１では、図４のステップＳ４０５で＜Ｔｅｘｔ＞要素のグリフＩＤの属性データを記述する際、各文字の幅情報と行高さとに基づいて、各文字に対応させるグリフを決定していたが、これに限るものではない。

【０１１８】

例えば、ステップＳ２０５の文字認識処理時に取得した各文字画像の位置情報を用いて、注目文字の左端から次の文字の左端までの間隔（文字送り幅）を求め、当該間隔と文字行高さとの比率に基づいて、グリフＩＤを選択するようにしてもよい。なお、各文字行の最後の文字についてはその文字の文字幅を前記間隔として用いる。なお、このようにした場合、前記間隔の方が文字行高さよりも大きくなるときがあるので、幅が高さよりも大きい矩形のグリフ（例えば、幅が１１５２や１２８０などのグリフ）も用意しておけばよい。また、このようにした場合、図５のステップＳ５１６では、次の文字の描画開始位置Ｘは、Ｘ＝Ｘ＋Ｗｉで求められることになる。

【０１１９】

このように文字送り幅を基準としてグリフＩＤを選択するようにして生成した電子文書に対して、検索処理を行うと、図１６のように、キーワードに一致する文字列内の文字間も含めて強調処理されることになる。

【０１２０】

このように、実施形態２においても、透明テキストの描画位置（文字の位置座標など）を全ての文字に対して文字ごとに細かくしなくても、透明テキストの描画位置と文書画像内の各文字画像の位置とがほぼ合うようになる。また、格納するグリフの総数は限られた数（例えば、１０個）になるので、フォントデータのデータ量を抑えることができる。また、グリフの字形自体も単純化されて保存されているので、字形データ自体のデータ量も抑えることができる。

【０１２１】

＜実施形態３＞

上述した実施形態の図５のステップＳ５１６における次の文字の描画開始位置Ｘを決定するための別実施形態について述べる。

【０１２２】

上述した実施形態の図２のステップＳ２０５の文字認識処理で識別した各文字画像の位置の情報に基づいて、文字間距離の平均値を算出する。そして、図４のステップＳ４０５で＜Ｔｅｘｔ＞要素を記述する際に、当該文字領域における文字間距離の平均値を属性データ（ＡｖＣ）として記述しておく。そして、ステップＳ５１６では、その記述しておいた文字間距離平均値（ＡｖＣ）を用いて、次の文字の描画開始位置を決定するようにしてもよい。この場合、次の文字の描画開始位置は、Ｘ＝Ｘ＋Ｗｉ＋ＡｖＣとなる。

【０１２３】

＜実施形態４＞

また、上述した実施形態では、スキャン画像に対してＪＰＥＧ圧縮等を行った全面イメージを＜Ｉｍａｇｅ＞要素に記述し、透明テキストを＜Ｔｅｘｔ＞要素に記述した電子文書を生成することとしたが、これに限るものではない。

【０１２４】

例えば、＜Ｉｍａｇｅ＞要素に、スキャン画像全体をＪＰＥＧ圧縮したものを記述する代わりに、文字領域や図領域は色別に２値画像を作成してＭＭＲ圧縮したもの、それ以外の領域はＪＰＥＧ圧縮したものを格納するようにしてもよい。このように、文書画像に含まれる領域を解析して適応的に圧縮処理を行う方法は、例えば、特開平０７－２３６０６２号公報や特開２００２－０７７６３３号公報などに記載の方法を用いることができる。本発明の透明テキストを描画する際に用いるフォントデータのデータ量を抑える処理と、これらの画像圧縮処理とを組み合わせることで、更に高圧縮された電子文書を生成することが可能になる。

【０１２５】

また、全面イメージの代わりに、文字領域、図領域、表領域、写真領域などの部分領域だけを位置データとともに保存するようにしても構わない。

【０１２６】

＜実施形態５＞

上述した実施形態では、検索した結果に対応する個所を強調表示する際、画像の色（ｒ，ｇ，ｂ）を反転することにより強調表示したが、使用する色はこれに限るものではない。例えば、検索結果を特定させるための予め決めた色（例えば黄色）を、半透明（例えばアルファチャネル１２８）で描画させるようにしてもよい。また、文字色（Ｃｒ，Ｃｇ，Ｃｂ）を利用して、強調色を決めるようにしてもよい。

【０１２７】

＜実施形態６＞

また、上述した実施形態では、図３及び図５で説明したように、検索を行う際は、キーワードに一致する文字列を文書の先頭から順に検索していき、最初に検索された文字列を強調表示した。そして、「次を検索」の指示があれば、順次、次に一致する文字列を検索して強調表示するように構成した。このように、上述した実施形態では、検索キーワードに一致する文字列を先頭から順に検索をおこない、検索キーワードがヒットするごとに順次強調表示を行っていたが、これに限るものではない。例えば、電子文書内に含まれる全ての文字列について、検索キーワードと比較を行い、全ての一致する文字列を特定し、そのキーワードに一致した全ての文字列を同時に強調表示するような構成にしてもよい。

【０１２８】

＜その他の実施形態＞

なお、本発明の目的は、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コード（コンピュータプログラム）を記憶した、コンピュータ読取可能な記憶媒体を、システムあるいは装置に供給することによっても達成される。また、システムあるいは装置のコンピュータ（またはＣＰＵやＭＰＵ）が記憶媒体に格納されたプログラムコードを読

出し実行することによっても達成される。

【０１２９】

本発明のコンピュータプログラムは、上述したフローチャートに記載した各ステップを装置に実行させることになる。言い換えると、このコンピュータプログラムは、フローチャートの各ステップに対応する各処理部（各処理手段）として、コンピュータを機能させるためのプログラムである。この場合、コンピュータ可読記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【０１３０】

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、フレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、ＣＤ－ＲＯＭ、ＣＤ－Ｒ、不揮発性のメモリカード、ＲＯＭなどを用いることができる。

【０１３１】

また、プログラムコードの指示に基づき、コンピュータ上で稼働しているＯＳ（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態が実現される場合も本発明に含まれることは言うまでもない。

【０１３２】

また、一方、上述した実施形態１，２では、ＣＰＵがメモリやハードディスク、表示デバイス等と協働して各フローチャートの各ステップを実行する形態について説明した。本発明は、上述した構成に限るものではなく、各フローチャートで説明した各ステップの処理の一部または全部を、ＣＰＵの代わりに専用の電子回路で構成するようにしても構わない。

【図面の簡単な説明】

【０１３３】

【図１】実施形態１の構成例を表すブロック図

【図２】実施形態１の電子文書生成処理の例を表すフローチャート

【図３】実施形態１の電子文書検索・閲覧処理の例を表すフローチャート

【図４】図２のステップＳ２０８でおこなわれる電子文書データ生成処理の詳細を表すフローチャート

【図５】図３のステップＳ３０６でおこなわれるページの描画処理の詳細を表すフローチャート

【図６】実施形態１により生成される電子文書の例

【図７】処理対象のページ画像の例

【図８】領域分割処理結果の例

【図９】生成される領域データの例

【図１０】文字認識処理時に文字画像を抽出する際の処理を示す例

【図１１】文字認識結果で生成される文字コード列データの例

【図１２】文字コード配列テーブルの例

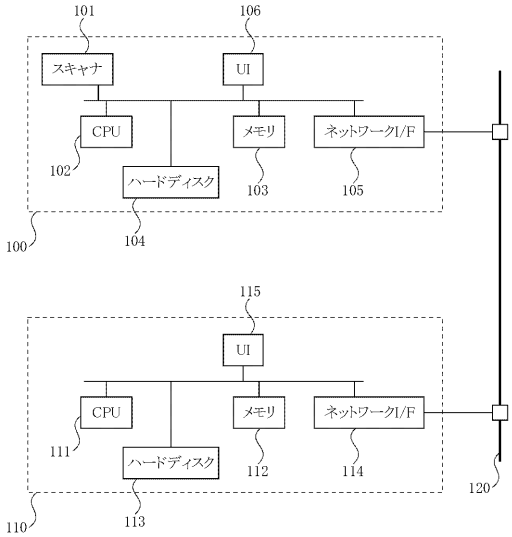
【図１３】検索結果が強調表示されたページ表示の例

【図１４】別の強調表示処理で検索結果が強調表示されたページ表示の例

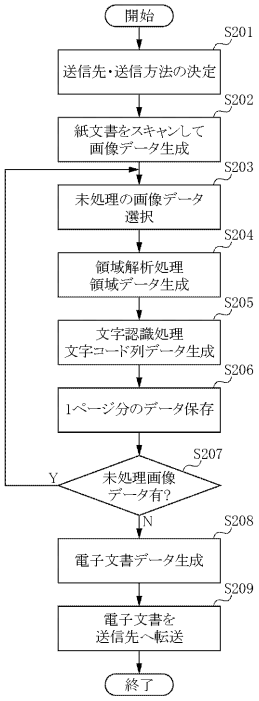
【図１５】グリフＩＤ決定処理を示すフローチャートの例

【図１６】実施形態２において、検索結果が強調表示されたページ表示の例

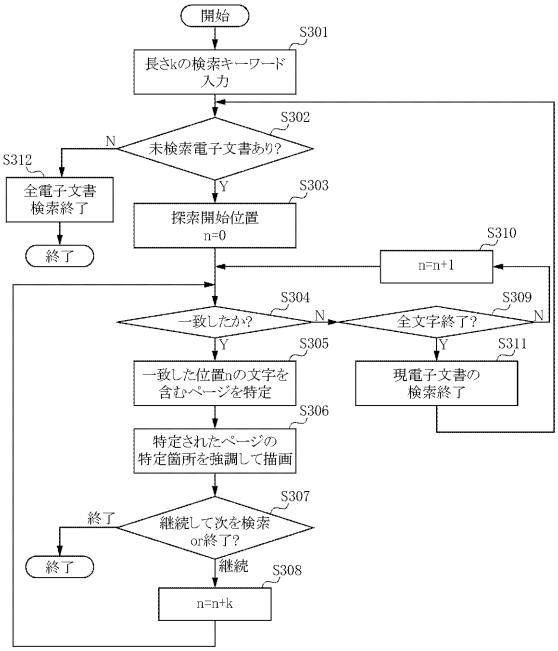
【図１】


101
106
スキャナ
UI
CPU
メモリ
ネットワークI/F
102
ハードディスク
103
105
100
104
115
UI
CPU
メモリ
ネットワークI/F
111
ハードディスク
112
114
120
110
113

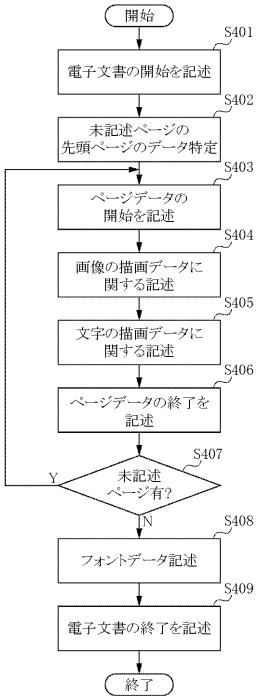
【図２】


開始
S201
送信先・送信方法の決定
S202
紙文書をスキャンして
画像データ生成
S203
未処理の画像データ
選択
S204
領域解析処理
領域データ生成
S205
文字認識処理
文字コード列データ生成
S206
1ページ分のデータ保存
S207
未処理画像
データ有?
Y
N
S208
電子文書データ生成
S209
電子文書を
送信先へ転送
終了

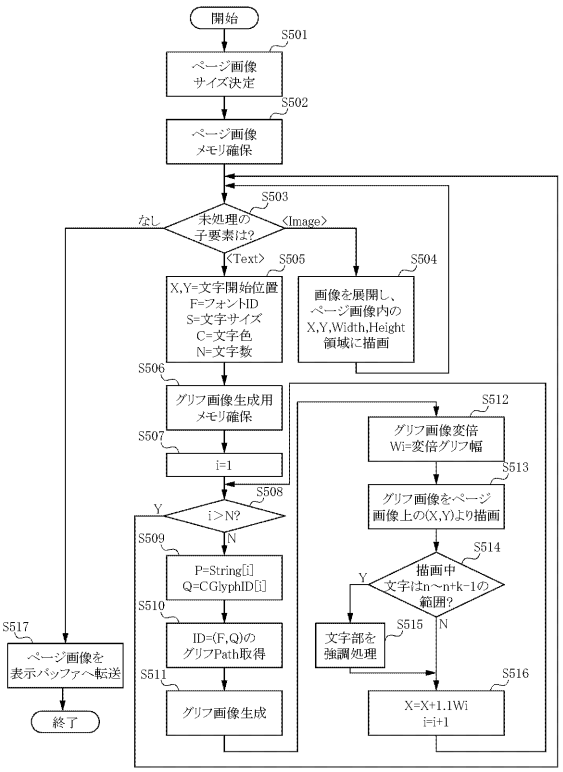
【図３】


開始
S301
長さkの検索キーワード
入力
S302
未検索電子文書あり?
N
Y
S312
全電子文書
検索終了
終了
S303
探索開始位置
n=0
S310
n=n+1
S304
一致したか?
N
Y
S309
全文字終了?
N
Y
S305
一致した位置nの文字を
含むページを特定
S311
現電子文書の
検索終了
S306
特定されたページの
特定箇所を強調して描画
S307
継続して次を検索
or終了?
終了
継続
終了
S308
n=n+k

【図４】


開始
S401
電子文書の開始を記述
S402
未記述ページの
先頭ページのデータ特定
S403
ページデータの
開始を記述
S404
画像の描画データに
関する記述
S405
文字の描画データに
関する記述
S406
ページデータの終了を
記述
S407
未記述
ページ有?
Y
N
S408
フォントデータ記述
S409
電子文書の終了を記述
終了

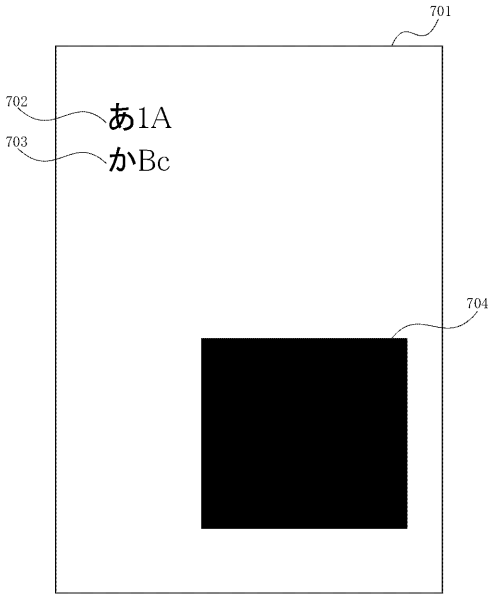
【図５】



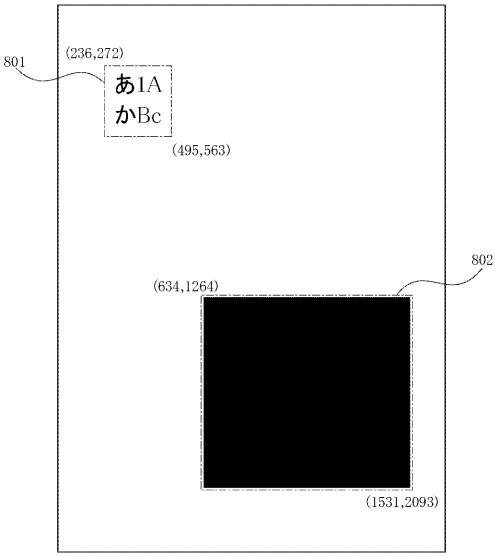
【図６】

```
601 <?xml version="1.0"?>
    <Document>
602   <Page Width="1680" Height="2376" Dpi="200" Number="1">
603     <Image X="0" Y="0" Width="1680" Height="2376"
              Data="ACAf49sw89q34rs6dbc23cfasx....a3E52zAD"/>
604     <Text X="236" Y="272" Font="Font01" Size="97"
              Color="0,0,0,255" String="0x2422,0x2332,0x2341"
              CGlyphId="1,6,3"/>
605     <Text X="236" Y="472" Font="Font01" Size="92"
              Color="0,0,0,255" String="0x242b,0x2342,0x2363"
              CGlyphId="0,2,5"/>
606   </Page>

607   <Page Width="1680" Height="2376" Dpi="200" Number="2">
608     <Image X="0" Y="0" Width="1680" Height="2376"
              Data="sa8OAN2qx7sa23Basd2x....aAqw8D2dD]"/>
609     <Text X="136" Y="872" Font="Font01" Size="92"
              Color="0,0,0,255" String="0x242b,0x242d,0x242f"
              CGlyphId="1,1,6"/>
610   </Page>

611   <Font Id="Font01">
        <Glyph Id="0" Path="M0,0 V-1024 H1024 V1024 f"/>
        <Glyph Id="1" Path="M0,0 V-1024 H896 V1024 f"/>
        <Glyph Id="2" Path="M0,0 V-1024 H768 V1024 f"/>
        <Glyph Id="3" Path="M0,0 V-1024 H640 V1024 f"/>
        <Glyph Id="4" Path="M0,0 V-1024 H512 V1024 f"/>
        <Glyph Id="5" Path="M0,0 V-1024 H384 V1024 f"/>
        <Glyph Id="6" Path="M0,0 V-1024 H256 V1024 f"/>
        <Glyph Id="7" Path="M0,0 V-1024 H128 V1024 f"/>
      </Font>
612 </Document>
```

【図７】



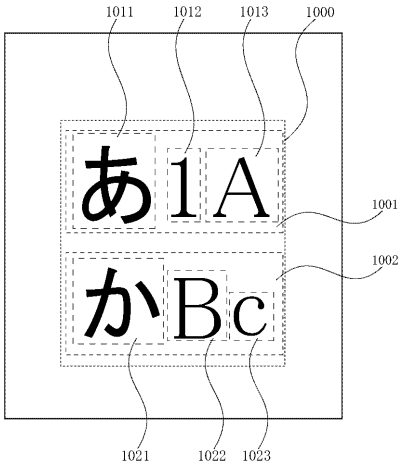
【図８】



【図９】

領域	x	y	width	height	種別
1	236	272	260	292	テキスト
2	634	1264	898	830	写真

【図１０】



【図１１】

1101

文字	x	y	width	height	文字幅/行高さ	コード
1	236	272	80	97	0.82	0x2422
2	339	284	14	75	0.14	0x2332
3	401	274	55	75	0.57	0x2341

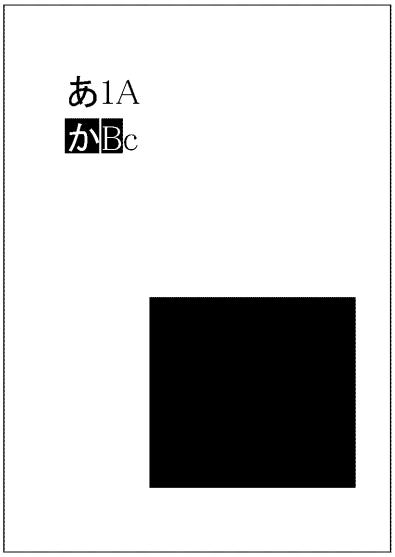
1102

文字	x	y	width	height	文字幅/行高さ	コード
1	236	474	92	85	1.00	0x242b
2	348	472	58	92	0.63	0x2342
3	428	472	28	89	0.30	0x2363

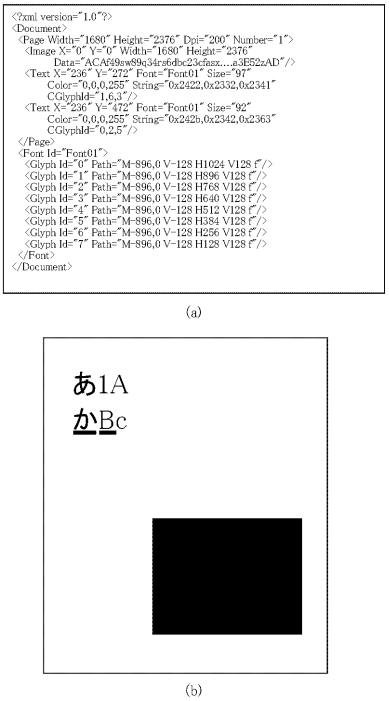
【図１２】

No.	文字コード	記述位置
0	0x2422	1093
1	0x2332	1100
2	0x2341	1107
3	0x242b	1250
4	0x2342	1257
5	0x2363	1264
6	0x242b	2601
7	0x242d	2608
8	0x242f	2615

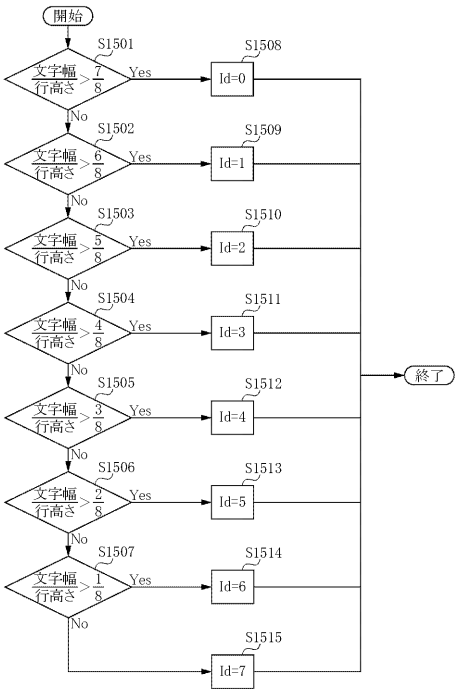
【図１３】



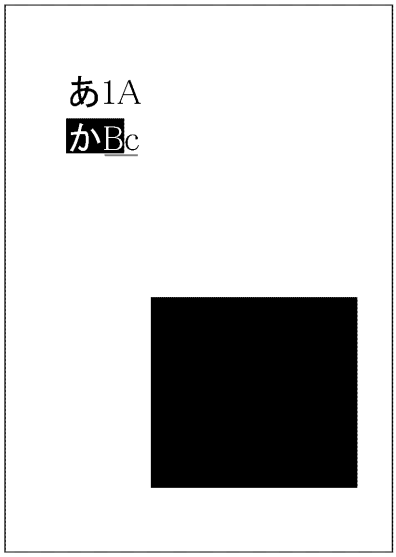
【図１４】



【図１５】



【図１６】



フロントページの続き

(72)発明者　山▲崎▼　妙子
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　佐藤　実

(56)参考文献　特開２００５－２５９０１７（ＪＰ，Ａ）
特開平１１－２３２２７６（ＪＰ，Ａ）
特開平０７－２７１８２９（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｋ　　９／００
Ｇ０６Ｆ　１７／３０