

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5376795号
(P5376795)

(45) 発行日 平成25年12月25日(2013.12.25)

(24) 登録日 平成25年10月4日(2013.10.4)

(51) Int. Cl. F 1
G06F 17/21 (2006.01) G06F 17/21 562D
G06T 1/00 (2006.01) G06T 1/00 200D

請求項の数 12 (全 26 頁)

(21) 出願番号	特願2007-321283 (P2007-321283)	(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
(22) 出願日	平成19年12月12日(2007.12.12)	(74) 代理人	110001243 特許業務法人 谷・阿部特許事務所
(65) 公開番号	特開2009-146064 (P2009-146064A)	(74) 代理人	100077481 弁理士 谷 義一
(43) 公開日	平成21年7月2日(2009.7.2)	(74) 代理人	100088915 弁理士 阿部 和夫
審査請求日	平成22年12月9日(2010.12.9)	(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
		(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

最終頁に続く

(54) 【発明の名称】 画像処理装置、画像処理方法、そのプログラム及び記憶媒体

(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、

前記文書画像と、前記文字認識手段で得た複数の文字コードと、複数の異なる文字コードの描画において共通で利用する字形データと、どの字形データを使用すべきかを各アプリケーションで判断するための判断基準の情報とを格納した電子文書を生成する生成手段と、

を有する画像処理装置であって、

前記複数の異なる文字コードの描画において共通で利用する字形データとして、複数種類の字形データが前記電子文書に格納され、

前記判断基準の情報と、前記複数種類の字形データそれぞれの属性データとして各字形データの形状の特徴を記述した情報であり、

前記電子文書に格納されている前記文書画像と前記文字コードとを描画することによって前記電子文書の閲覧処理を行うアプリケーションは、当該電子文書に格納されている前記複数種類の字形データの中から、前記複数種類の字形データそれぞれの属性データに記述されている各字形データの形状の特徴に基づいて、当該アプリケーションで強調表示するのに適した形状の特徴が前記属性データに記述されている字形データを自動的に1つ選択し、当該選択した1つの字形データを用いて当該電子文書に格納されている前記複数の異なる文字コードの描画を行う

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、

前記文書画像と、前記文字認識手段で得た複数の文字コードと、複数の異なる文字コードの描画において共通で利用する字形データと、どの字形データを使用すべきかを各アプリケーションで判断するための判断基準の情報とを格納した電子文書を生成する生成手段と、

を有する画像処理装置であって、

前記複数の異なる文字コードの描画において共通で利用する字形データとして、複数種類の字形データが前記電子文書に格納され、

前記判断基準の情報は、前記複数種類の字形データそれぞれの属性データとしてアプリケーションの名前または種類を記述した情報であり、

前記電子文書に格納されている前記文書画像と前記文字コードとを描画することによって前記電子文書の閲覧処理を行うアプリケーションは、当該電子文書に格納されている前記複数種類の字形データの中から、当該閲覧処理を行うアプリケーションの名前または種類が前記属性データに記述されている字形データを自動的に1つ選択し、当該選択した1つの字形データを用いて当該電子文書に格納されている前記複数の異なる文字コードの描画を行う

ことを特徴とする画像処理装置。

【請求項 3】

前記電子文書に格納される複数種類の字形データのうちの1つは、矩形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 または 2 に記載の画像処理装置。

【請求項 4】

前記電子文書に格納される複数種類の字形データのうちの1つは、下線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 または 2 に記載の画像処理装置。

【請求項 5】

前記電子文書に格納される複数種類の字形データのうちの1つは、波線、点線、三角、丸、四角形の形状のうちの少なくとも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 または 2 に記載の画像処理装置。

【請求項 6】

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の位置に対応す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 または 2 に記載の画像処理装置。

【請求項 7】

前記電子文書は、XMLフォーマット或いはXPSフォーマット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 または 2 に記載の画像処理装置。

【請求項 8】

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 または 2 に記載の画像処理装置。

【請求項 9】

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 8 に記載の画像処理装置。

【請求項 10】

前記アプリケーションは、前記生成された電子文書に格納されている複数の文字コードに対して、入力された検索キーワードに一致する部分を前記選択した1つの字形データを用いて強調表示させることを特徴とする請求項 1 乃至 9 のいずれか1項に記載の画像処理装置。

【請求項 11】

10

20

30

40

50

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、

生成手段が、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、複数の異なる文字コードの描画において共通で利用する字形データと、どの字形データを使用すべきかを各アプリケーションで判断するための判断基準の情報とを格納した電子文書を生成する生成ステップと、を有する画像処理方法であって、

前記複数の異なる文字コードの描画において共通で利用する字形データとして、複数種類の字形データが前記電子文書に格納され、

前記判断基準の情報は、前記複数種類の字形データそれぞれの属性データとして各字形データの形状の特徴を記述した情報であり、

10

前記電子文書に格納されている前記文書画像と前記文字コードとを描画することによって前記電子文書の閲覧処理を行うアプリケーションは、当該電子文書に格納されている前記複数種類の字形データの中から、前記複数種類の字形データそれぞれの属性データに記述されている各字形データの形状の特徴に基づいて、当該アプリケーションで強調表示するのに適した形状の特徴が前記属性データに記述されている字形データを自動的に1つ選択し、当該選択した1つの字形データを用いて当該電子文書に格納されている前記複数の異なる文字コードの描画を行う

ことを特徴とする画像処理方法。

【請求項12】

コンピュータを、請求項1乃至10のいずれか1項に記載の画像処理装置として機能させるためのプログラム。

20

【発明の詳細な説明】

【技術分野】

【0001】

本発明は紙文書を電子的に検索可能なデータへと変換する技術に関する。

【背景技術】

【0002】

近年、スキャナや、ハードディスク等の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電子化し、電子文書として保存する作業が進んでいる。特に紙文書をスキャンして画像データに変換するだけではなく、そこに書かれた文字情報を文字認識技術により読みとって、画像の付加情報として保存しておくことも一般に行われている。そのようにして保存された電子文書は、ユーザが元の文書に含まれていた文字列を検索キーワードとして入力することで、大量の保存文書群の中から所望の文書を高速に取り出すことが可能となっている。

30

【0003】

また、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。

40

【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。

50

【 0 0 0 5 】

【特許文献1】特開2000-322417号公報

【発明の開示】

【発明が解決しようとする課題】

【 0 0 0 6 】

P D F や S V G 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、フォントの種類を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。

10

【 0 0 0 7 】

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。

【 0 0 0 8 】

20

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。

【 0 0 0 9 】

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、X P S (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。

30

【 0 0 1 0 】

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。

【 0 0 1 1 】

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。

40

【 0 0 1 2 】

また、検索結果の強調表示を行う際に、文書を表示するビューアの特性によっては、検索結果の強調表示の仕方が異なる。すなわち、検索結果の強調表示により画像上の文字画像が見づらくなることがある。

【 0 0 1 3 】

かかる状況では、紙文書を電子的に検索可能な電子文書へと変換する処理において、以下の機能が求められる。すなわち、使用されるフォントデータを電子文書内部に保持するが当該電子文書のサイズは小さく抑えつつ、検索の強調表示時に視認性が確保できるよう

50

にすることが望まれる。

【課題を解決するための手段】

【0014】

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、前記文書画像と、前記文字認識手段で得た複数の文字コードと、複数の異なる文字コードの描画において共通で利用する字形データと、どの字形データを使用すべきかを各アプリケーションで判断するための判断基準の情報とを格納した電子文書を生成する生成手段と、を有する画像処理装置であって、前記複数の異なる文字コードの描画において共通で利用する字形データとして、複数種類の字形データが前記電子文書に格納され、前記判断基準の情報は、前記複数種類の字形データそれぞれの属性データとして各字形データの形状の特徴を記述した情報であり、前記電子文書に格納されている前記文書画像と前記文字コードとを描画することによって前記電子文書の閲覧処理を行うアプリケーションは、当該電子文書に格納されている前記複数種類の字形データの中から、前記複数種類の字形データそれぞれの属性データに記述されている各字形データの形状の特徴に基づいて、当該アプリケーションで強調表示するのに適した形状の特徴が前記属性データに記述されている字形データを自動的に1つ選択し、当該選択した1つの字形データを用いて当該電子文書に格納されている前記複数の異なる文字コードの描画を行うことを特徴とする。

10

【発明の効果】

20

【0017】

本発明によれば、紙文書をスキャンしたページ画像の描画記述と、ページ画像より抽出した文字を（透明色で）描画する記述を有する電子文書へと変換する。その際に、単純な字形からなるフォントデータを複数、内部に格納する。また、各フォントデータにおいて、1つの字形を複数の文字種に対して共通して利用させる。これにより、使用されるフォントデータを電子文書内部に保持するが、字形データは少なく済むので、当該電子文書のファイルサイズ（データ容量）は小さく抑えることができる。更に、いくつかのフォントデータを格納し、複数の字形を切り換えて描画することを可能としているので、異なるアプリケーションで検索の強調表示を行う場合であっても、視認性や操作性のよい表示が可能となる。

30

【発明を実施するための最良の形態】

【0018】

<実施形態1>

以下、本発明の一実施形態について図面を用いて説明する。

【0019】

図1は本発明を実施できる画像処理装置の構成を示すブロック図の一例である。

【0020】

100は紙文書を電子文書へと変換する画像処理装置であり、以下の各デバイスにより構成される。

【0021】

40

101は、読み取った紙文書の紙面情報を画像のデータに変換するスキャナである。102は、画像データを解析して検索可能な電子文書へと変換するプログラムなどを実行するCPUである。103のメモリおよび104のハードディスクは、上記プログラムによる電子文書への変換結果や途中のデータを保存するための記憶装置である。

【0022】

105は、上記プログラムによって生成されたデータを装置外に出力するネットワークI/Fである。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。

【0023】

110は100で作成された電子文書の検索・閲覧を行う画像処理装置であり、以下の

50

デバイスにより構成される。

【 0 0 2 4 】

1 1 1 は、電子文書のデータを解釈し検索や閲覧のための表示データを作成するプログラムや検索動作の制御を行うプログラムなどを実行する。1 1 2 のメモリや1 1 3 のハードディスクは、電子文書データの保存や、上記プログラムにより作成される表示データや処理途中のデータを保存するための記憶装置である。1 1 4 は、装置外で生成された電子文書を装置内に転送するためのネットワーク I / F である。1 1 5 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。

【 0 0 2 5 】

1 2 0 は画像処理装置 1 0 0 と画像処理装置 1 1 0 を電子的に接続するネットワークである。

【 0 0 2 6 】

次に、本実施形態 1 による処理の例を、図 2 および図 3 のフローチャートを用いて説明する。

【 0 0 2 7 】

図 2 は、画像処理装置 1 0 0 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 1 1 0 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。

【 0 0 2 8 】

はじめに、ステップ S 2 0 1 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 1 0 6 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。

【 0 0 2 9 】

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 S 2 0 2 では、スキャナ 1 0 1 が公知の光電変換技術を用いて紙文書をスキャンしてページ画像データ（文書画像）へと変換する。手動、あるいはオートドキュメントフィーダなどを用いて複数ページの文書が入力された場合は、スキャンされた紙文書は 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 1 0 3 に保存されるものとする。

【 0 0 3 0 】

図 7 にページ画像の例を示す。図 7 中のページ画像 7 0 1 には、「あいう」という文字列 7 0 2 と「かきく」という文字列 7 0 3 からなる文字画像、および写真 7 0 4 が存在する。なお、説明のために、写真 7 0 4 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 7 0 2 , 7 0 3 と、写真 7 0 4 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。

【 0 0 3 1 】

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 8 b i t でその階調を表現するカラー画像、白黒であれば 8 b i t で輝度を表現するグレースケール画像もしくは 1 b i t で白黒を表現する二値画像であるとする。

【 0 0 3 2 】

ステップ S 2 0 3 では、メモリ 1 0 3 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。

【 0 0 3 3 】

ステップ S 2 0 4 では、選択された画像から、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 1 0 3 に保存する。なお、この領域データには、該当領域の外接矩形の左上位置に対する画像内画素の x , y 方向の座標値 x , y、および該外接矩形のサイズ（幅・高さ）を表わす画素数の値 w i d t h , h e i g h t、さら

10

20

30

40

50

に、テキスト、写真などの領域の種別が含まれるものとする。

【0034】

上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。

【0035】

ここで、図8および図9に、図7に示したページ画像701にたいする領域解析処理例を示す。図8中の801がテキスト領域、802が写真領域と判定された領域である。図9がその領域解析処理で得られた領域データの例である。

10

【0036】

次に、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、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、および当該各文字画像の外接矩形情報（外接矩形の左上座標 x, y とその幅・高さの情報 $width, height$ ）が含まれるものとする。

【0037】

ここで、上記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。

20

【0038】

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。

30

【0039】

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。

【0040】

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。

40

【0041】

なお上記説明は一例であって、他の公知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。

【0042】

図10および図11に、図8で示したテキスト領域801に対する文字認識処理例を示している。

50

【 0 0 4 3 】

図 1 0 中のテキスト領域 1 0 0 0 から文字行 1 0 0 1 と 1 0 0 2 が先ず切り出され、さらに文字行 1 0 0 1 内から 1 0 1 1 , 1 0 1 2 , 1 0 1 3 の 3 文字が切り出される。そして、それぞれの文字が認識され、その結果、各文字に対応する文字コードが得られて、図 1 1 の 1 1 0 1 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 1 0 0 2 内から切り出された 1 0 2 1 , 1 0 2 2 , 1 0 2 3 の 3 文字にも文字認識処理が実行され、図 1 1 中の 1 1 0 2 の文字コード列データが生成される。

【 0 0 4 4 】

次いで、ステップ S 2 0 6 では、処理対象となっているページ画像データ、領域データおよび文字コード列データとを関連付けて、メモリ 1 0 3 もしくはハードディスク 1 0 4 に保存する。

10

【 0 0 4 5 】

ステップ S 2 0 7 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 S 2 0 3 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 S 2 0 8 に進む。

【 0 0 4 6 】

ステップ S 2 0 8 では、メモリ 1 0 3 あるいはハードディスク 1 0 4 に保存された全ページ分のデータをページ順（処理順）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。

【 0 0 4 7 】

上記ステップ S 2 0 8 において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。そのようなデータ保持条件を満たすデータフォーマットとしては P D F 形式、 S V G 形式など様々な公知例が存在する。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、 X P S などがある。以下では、 X M L 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。もちろん、既存の、字形を埋め込む形式の X P S 、 P D F / A などの電子文書フォーマットを用いてもよい。

20

30

【 0 0 4 8 】

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 Z I P 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、 X P S ）でもよい。

【 0 0 4 9 】

以下に、ステップ S 2 0 8 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。

40

【 0 0 5 0 】

はじめに、ステップ S 4 0 1 では電子文書の開始を表わす記述を生成する。

【 0 0 5 1 】

本説明のページデータ記述フォーマット仕様では、 < D o c u m e n t > という要素が電子文書の開始タグを表し、その終了である < / D o c u m e n t > までに挟まれた範囲の X M L 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 6 0 1 が電子文書の開始タグ、 6 1 3 が終了タグを表す。

【 0 0 5 2 】

ステップ S 4 0 2 では未記述のページの中から先頭ページ用のデータを特定して処理対

50

象とする。

【0053】

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。

【0054】

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 = "1680"、高さHeight = "2376"および解像度Dpi = "200"、ページ番号Number = "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。

【0055】

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記述(画像描画記述)を生成する(画像描画記述生成)。

【0056】

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。属性Data内に記述されるのは公知方法で圧縮された画像データ、例えばカラーやグレースケールの画像データの場合はJPEG圧縮を、二値の画像データの場合はMMR圧縮をしたコード列であるとする。

【0057】

図6の記述例では、図2のステップS203で選ばれた文書1ページ目のスキャン画像がページ面にわたって描画されるように、タグ603では、X = "0"、Y = "0"、Width = "1680"、Height = "2376"が記述されている。さらに画像をJPEG圧縮したコード列をテキスト化した文字列を属性Dataの値とした<Image>要素603が記述されている(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)。

【0058】

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(文字描画記述)を生成する(文字描画記述生成)。

【0059】

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<Text>要素がそれぞれ縦または横の1行分の文字の描画データを表わす。<Text>要素内に記述される属性データは以下のとおりである。

- ・文字列の縦書き/横書きを示す属性Direction(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)
- ・文字の開始位置の座標を指定する属性X, Y
- ・文字を描画する際に適用されるフォントデータのIDを指定する属性Font
- ・フォントサイズを指定する属性Size
- ・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ンネル値の4値の組で指定する属性Color
- ・文字列の内容(文字コード列)を指定する属性String
- ・String内の各文字から次の文字までの送り幅を指定する属性CWidth
- ・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を指定する属性CGlyphId

10

20

30

40

50

【 0 0 6 0 】

ここで、< T e x t >要素を構成する文字列は、図2のステップS 2 0 5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である。なお、属性F o n tが未定義である場合、デフォルトのフォントIDに対応する字形が、全文字共通の字形として用いられる。

【 0 0 6 1 】

図6の記述例では、2つの< T e x t >タグ6 0 4および6 0 5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1 1 0 1および1 1 0 2それぞれに対応する記述である。

【 0 0 6 2 】

例えば図11中1 1 0 1の3文字の横書き文字列「あいう」に対応する< T e x t >要素記述6 0 4においては、以下のとおり、各属性値が記述されている。

【 0 0 6 3 】

・属性X , Yは、3文字分の外接矩形の左上座標としてX = “ 2 3 6 ”、Y = “ 2 7 2 ” が指定されている。

【 0 0 6 4 】

・フォントデータIDの属性F o n tは未定義である。

【 0 0 6 5 】

・フォントサイズの属性S i z eは、行内文字の高さから類推して“ 9 7 ”ピクセルが指定されている。

【 0 0 6 6 】

・属性D i r e c t i o nは横書き“ H o r i z o n t a l ”が指定されている。

【 0 0 6 7 】

・描画時文字色の属性C o l o rは、R成分値 = G成分値 = B成分値 = 0、アルファチャンネル = 2 5 5を意味する“ 0 , 0 , 0 , 2 5 5 ”が指定されている（つまり、透明色が指定されている）。

【 0 0 6 8 】

・文字列内容（各文字に対応する文字コードの列）を指定する属性S t r i n gは、“ 0 x 2 4 2 2 , 0 x 2 4 2 4 , 0 x 2 4 2 6 ”が指定されている。

【 0 0 6 9 】

・各文字の送り幅を指定する属性C W i d t hは、左の2文字は右隣文字との左端の座標差、最後の文字は自らの文字幅に相当する、“ 1 0 4 , 9 6 , 5 9 ”が指定されている。

【 0 0 7 0 】

・各文字の字形データであるグリフのIDを指定する属性C G l y p h I dには、通常、各文字の字形データに合わせたグリフのIDが指定される。しかしながら、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、字形が異なるものであっても、ユーザの視覚には見えていない。そこで、本実施形態では、異なる文字（文字種）であっても、同一のグリフIDを指定することにより字形データ（フォントデータ）が少なく済むようにしている。したがって、図6の例では、属性C G l y p h I dには、“ 0 , 0 , 0 ”の同じ属性値が記述されている。また、このグリフIDで指定される字形は、簡単な形状（例えば矩形）でよい。なお、グリフの形状の詳細については、後述する。

【 0 0 7 1 】

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、ピクセル高さや画像解像度から、画素数ではなくポイント数等の値で記述されてもよい。あるいは、本記述で描画される文字には透明色が指定され、ユーザには見えていないので、描画される文字列が、対応する文字画像の真上に重ならなくても構わない。例えば、対応する文字画像の下端部に、透明な文字列が描画されるようにしてもよい。例えば、図6の6 0 4の例であれば、X = “ 2 3 6 ”、Y = “ 3 6 8 ”

10

20

30

40

50

”、Size = “ 1 0 ” とすれば、文字画像の下端に高さの低い透明な文字列が描画されることになる。このとき、この描画される透明文字列のサイズ（高さ）は、文字画像よりも小さい所定のサイズ（例えば 1 0 ）としている。

【 0 0 7 2 】

描画される透明文字列は、後に、検索キーワードで検索を行う際に用いられ、検索キーワードに一致する文字列は強調表示（例えば、色が変わられて表示）される。透明文字列は、対応する文字画像の位置にほぼ対応する位置に描画されているので、検索時には透明文字列を使って検索しているのであるが、ユーザにとっては、あたかも文字画像が検索されているかのように見えることになる。したがって、このような検索時の文字の強調の用途に用いるのであれば、透明文字列を対応する文字画像の下端部に描画したとしても、検索時には、対応する文字画像にアンダーラインが引かれたかのように強調表示されて特定できるので問題ない。なお、透明文字列の描画位置は、下端に限るものではなく、文字画像の下半分や上半分というような位置に描画されるよう記述しても構わない。

10

【 0 0 7 3 】

次に、ステップ S 4 0 6 では、ページ記述の終了を示す < / P a g e > を記述する。

【 0 0 7 4 】

ステップ S 4 0 7 では未記述のページの有無を判定し、未記述のページがある場合はステップ S 4 0 3 から繰り返し、ない場合はステップ S 4 0 8 に進む。図 6 の記述例では 2 ページ目に対してもステップ S 4 0 4 ~ S 4 0 6 の処理が行われ、6 0 7 ~ 6 1 0 の部分が記述されることになる。

20

【 0 0 7 5 】

ステップ S 4 0 8 では、この電子文書で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の記述（字形データ記述）を生成する（字形データ記述生成）。本説明のページデータ記述フォーマット仕様では、< F o n t > と < / F o n t > に挟まれる範囲にフォントデータに含まれるグリフの一つ一つが < G l y p h > 要素として記述される。< F o n t > 要素には、当該フォントの種類を示す属性 I D が含まれる。また、< G l y p h > 要素には、グリフの種類を示す属性 I D と、その I D に対応するグリフ（字形）を示す属性 P a t h とが含まれる。< G l y p h > 要素の属性 P a t h は、左下を原点とする 1 0 2 4 × 1 0 2 4 描画矩形単位内で、直線あるいは曲線関数を用いてグリフを表現する記述である。

30

【 0 0 7 6 】

図 6 の記述例では、6 1 1 および 6 1 2 の < F o n t > 要素にて、6 1 1 ではフォント I d = “ F o n t 0 1 ”、6 1 2 ではフォント I d = “ F o n t 0 2 ” のフォントがそれぞれ定義されている。それぞれの内容に I d = “ 0 ” を持つグリフが一種類定義されている。6 1 1 の “ F o n t 0 1 ” のグリフの字形を表わす P a t h 属性の “ M 0 , 0 V - 1 0 2 4 H 1 0 2 4 V 1 0 2 4 f ” の意味は、以下のとおりである。

「原点（0，0）に M O V E ，上方向に 1 0 2 4 単位縦線を描画、右方向に 1 0 2 4 単位横線を描画、下方向に 1 0 2 4 単位縦線を描画、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。」

すなわち、1 0 2 4 × 1 0 2 4 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。

40

【 0 0 7 7 】

また 6 1 2 の “ F o n t 0 2 ” のグリフの字形を表わす P a t h 属性の “ M 0 , 0 V - 6 4 H 1 0 2 4 V 6 4 f ” の意味は、以下のとおりである。

「原点（0，0）に M O V E ，上方向に 6 4 単位縦線を描画、右方向に 1 0 2 4 単位横線を描画、下方向に 6 4 単位縦線を描画、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。」

すなわち、描画矩形単位内下部の 1 0 2 4 × 6 4 の領域を塗りつぶす水平方向直線状のグリフを表現する記述となっている。

【 0 0 7 8 】

50

なお、図6の要素611および612の記述は一例であって、それぞれ垂直方向直線や波線や点線や三角や丸や四角形などの別の単純な字形を定義してもよい。

【0079】

次に、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。

【0080】

ここで図2に戻る。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で指定された送信方法で、指定された送信先である画像処理装置110へと転送する。転送処理は公知技術を用いるものとして説明は省略する。

10

【0081】

以上のようにして、転送された電子文書は画像処理装置110がネットワークインタフェース114を通して受信し、ハードディスク114へと蓄積する。

【0082】

ここで、蓄積する電子文書をハードディスク内部で特定するためのファイル名などのID情報は任意のものでよく、本説明では例として受信時刻に関連する文字列を付与するものとする。他にも、例えば重複しない番号を選択して自動付与したり、あるいは画像処理装置100で生成時にユーザが指定する情報としてファイル名を予めユーザが入力しておくなどの方法があるが、本発明の本質とは異なる処理であるため詳細な説明は省略する。

20

【0083】

次に、図1の画像処理装置110により電子文書を検索・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。

【0084】

ステップS301では、Font属性未定義の場合に使用されるデフォルトのフォントIDを電子文書が保持しているフォントIDの一覧からUI115を用いてユーザに選択させる。図16は検索対象の電子文書が所持しているフォントの一覧と、選択中のフォントのプレビュー画像(図中、黒四角部分はFont01のグリフを示している)を表示した選択画面UIの一例を示したものである。このようなUIを用いてユーザはデフォルトのフォントIDを選択できる。なお、図3中のステップS301のフォントID指定処理を行う順序は一例であり、ステップS307より前であればどこにあってもよい。

30

【0085】

ステップS302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、当該電子文書のテキストに含まれているとユーザにより考えられた検索キーワードがUI115より入力される。ここで入力された文字列の長さをkとする。

【0086】

ステップS303では、画像処理装置110のハードディスク114内にあるすべ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を判断する。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S304に進む。未検索の電子文書がなければS313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。

40

【0087】

ステップS304では、S303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、検索キーワード文字列の探索を行う準備を行う。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。

【0088】

ここで、ステップS304の処理例の詳細を以下に説明する。

50

【 0 0 8 9 】

電子文書データをXMLパーサでパースしていく段階で、<Text>要素が表われたとき属性Stringに記述されている文字コード列を取得する。そして、その中のString属性中に記載された1文字ずつ、その文字コードと、その文字コード値が電子文書データ内で記述されている位置との組を、文字コード並び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、当該文字コードを記述するキャラクタの先頭が、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ことを示す値である。

【 0 0 9 0 】

ここで、理解を容易にするため図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。この例では、図6中の<Text>要素604の属性Stringには3つの文字コード“0x2422”、“0x2424”、“0x2426”が記述されている。ここでは、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<Text>要素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。

10

【 0 0 9 1 】

次に、ステップS305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。一致する部分を抽出した場合、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S306に進む。

20

【 0 0 9 2 】

一方、ステップS305で一致しないと判断した場合は、ステップS310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S312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S311に進んで、変数nを1インクリメントして、ステップS305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S310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、 $n \geq (N - k)$ ならば探索終了と判断すればよい。

30

【 0 0 9 3 】

図12の文字コード配列テーブルの例に対し、例えばキーワード文字「かき」の文字コード列“0x242b, 0x242d”を先頭から走査して一致する部分を探した場合、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。また、後述するS307で更に検索を継続して次の一致文字列を検索した場合は位置 $n = 6$ が抽出される。なお、この文字コードと記述位置を対にした文字コード並びテーブルを用いたステップS303～S305の処理はあくまで一例であって、他の方法を用いてもよい。

【 0 0 9 4 】

次いで、ステップS306では、一致した文字列番号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。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S306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S304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S305の一致文字列の検出方法や、ステップS306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。

40

【 0 0 9 5 】

50

ステップS307では、ステップS306で決定されたページ内に含まれる描画記述に従って、ページの描画結果をUI115に表示する（電子文書表示）。このとき、文字列番号（No.）が $n \sim n+k-1$ 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。

【0096】

ここで、ステップS307で実施されるページの描画処理の詳細を、図5のフローチャートに従って以下に説明する。

【0097】

ステップS501では、<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。

10

【0098】

ステップS502では、ページ画像の画素情報が格納できる量のメモリ領域を確保する。

【0099】

ステップS503では、<Page>要素の子要素の中で未処理の要素を先頭から順に1つ抽出し、当該抽出した未処理要素の種類を判別する。未処理要素が<Image>であると判別した場合ステップS504に進む。一方、未処理要素が<Text>であると判別した場合ステップS505に進む。当該<Page>要素の全ての子要素が既にすべての要素が処理されている場合はステップS517へ進む。

20

【0100】

ステップS504では、<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属性が表わすページ画像内の描画矩形領域いっぱい収まるように当該展開されたイメージを変倍する。そして、上記ステップS502で取得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。

【0101】

一方、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）、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。なお、図6の例のようにフォントIDが定義されていない場合は、前述のS301で指定されたデフォルトのフォントIDを用いる。

30

【0102】

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

【0103】

ステップS507では、処理中文字カウンタ*i*を1に初期化する。

【0104】

ステップS508では、 $i > N$ であるか否かの判断を行い、 $i \leq N$ ならステップS509に進み、 $i > N$ なら当該<text>要素の処理は終了してステップS503に戻る。

40

【0105】

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。

【0106】

ステップS510では、電子文書の中からFontId = Fの要素記述を探し出し、更に、その要素記述の子要素の中でグリフId = Qの<Glyph>要素からPath属性を取得する。

【0107】

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対してグリフの二値画像を生成する

50

。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。

【0108】

ステップS512では、グリフの二値画像を、文字サイズの属性値(S)に則した大きさの矩形サイズになるよう変倍する。

【0109】

ステップS513では、ページ画像メモリの位置(X, Y)から始まる矩形領域に、変倍されたグリフの二値画像の情報を描画する。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。このとき、ページ画像の対象領域内の各画素値(r, g, b)は、二値画像中の対応する画素の値が0か1かによって以下の(r', g', b')へと各々変化する。

・グリフ二値画像の画素値が0に対応する画素の場合：

$$(r', g', b') = (r, g, b)$$

・グリフ二値画像の画素値が1に対応する画素の場合：

$$(r', g', b') = (F(r, Cr), F(g, Cg), F(b, Cb))$$

ここで、 $F(r, Cr) = (r \times A + Cr \times (255 - A)) / 255$ 、 $F(g, Cg) = (g \times A + Cg \times (255 - A)) / 255$ 、 $F(b, Cb) = (b \times A + Cb \times (255 - A)) / 255$ とする。また、Aは文字色Cのアルファチャネル値、Cr, Cg, Cbは文字色CのそれぞれRGB値である。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。

【0110】

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。

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、対応するグリフ二値画像の画素値が0である画素はそのままに、対応するグリフ二値画像の画素値が1である画素に対しては各画素値(r, g, b)を以下の(r', g', b')へと各々変化させる

$$(r', g', b') = (G(r), G(g), G(b))$$

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

【0111】

なお、上記強調処理は一例であり、例えば強調する幅をグリフ二値画像の幅ではなく、各文字の送り幅を指定する属性CWidthの値を用いて、連続する文字が間隔無しに塗りつぶされるようにしてもよい。

【0112】

一方、ステップS516ではCWidth属性のi番目に記述されたこの文字の送り幅をXに加算するとともにiに1を加算して(i=i+1)、ステップS503からの処理を繰り返す。ステップS503の段階で未処理の子要素がなくなると、ステップS517へ移る。

【0113】

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>要素記述の描画結果となっている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。

【0114】

10

20

30

40

50

次いで、図5のフローチャートが示す、図3のステップS307の処理を、図6の電子文書中1ページ目の描画記述に従って実行した場合の例を説明する。

【0115】

ステップS501の処理により、図6中の1ページ目の<Page>要素602の属性値Width = "1680"、Height = "2376"から、ページの画像サイズは1680×2376ピクセルと決定される。

【0116】

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。

【0117】

ステップS504の処理により、図6中の<Image>要素603のData属性値に記述された圧縮コードが展開されて画像データとなり、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々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。

【0118】

ステップS505の処理により、図6中の<Text>要素604から、X = "236"、Y = "272"、文字数N = 3、文字サイズ = "97"、文字色 = "0, 0, 0, 255"が得られる。<Text>要素604ではFont要素が定義されていない為、S301で指定したデフォルトの文字コードが指定される。ここでは、S301でFontID = "Font01"が指定されていたとする。

【0119】

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

【0120】

ステップS511で生成されるグリフの二値画像は、デフォルトのFontID = "Font01"が指定されている為、図6中の要素611内にある、Id = "0"の<Glyph>要素に記述されたPath属性に基づいて作成される。具体的にはPath属性の記述に従って、1024×1024ピクセルのGlyph画像領域すべてを1で塗りつぶした画像となる。

【0121】

なお、図6の電子文書に記載された<Text>要素604および605内の文字のCGlyphIdはすべて"0"であるため、結果的にすべての文字に対してステップS511の処理結果は等しくなる。したがって、ステップS511で生成したグリフ画像はメモリに一時保存しておき、他の文字を描画する際、その一時保存されているグリフ画像を流用するようにしてもよい。

【0122】

ステップS512では、グリフの文字画像が文字サイズ = "97"によって97×97ピクセルに変倍される。

【0123】

ステップS513では、ページ画像の位置(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。しかし図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、グリフの二値画像の対応する画素値は1でも常に(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。

【0124】

ステップS514では、図6中の<Text>要素604内の1番目の文字が、図3中のステップS305で得られた位置n ~ n + k - 1の範囲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。

10

20

30

40

50

【 0 1 2 5 】

ここでは、例えば図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中のステップS305でキーワードと一致すると判断された文字列の位置が3~4の範囲であったとする。このとき、図6中の<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図に示すように1093である。これは、文字コード配列テーブルの3~4の範囲内の文字の記述位置のいずれとも一致しないので、ステップS516を経て次の文字へと処理が進む。

【 0 1 2 6 】

その後、処理が進んで図6中の<Text>要素605内の1番目が示す文字の処理に於いては、ステップS514において、文字コード配列テーブルの3~4の範囲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。

10

【 0 1 2 7 】

この文字に対し、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92x92の領域内の対応するグリフ二値画像の画素値が1である画素の各画素値(r, g, b)を、(G(r), G(g), G(b))へと変化させる。

【 0 1 2 8 】

以上のようにして、全ての<Text>を描画した後のページ画像は図13に示すようになる。すなわち、ステップS305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る。一方、残りの文字に対応する領域は、<Image>要素を描画した画像データのままとする。

20

【 0 1 2 9 】

したがって、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S307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。

【 0 1 3 0 】

一方、ビューア・アプリケーションの種類によっては、強調表示する際の表示方法が異なる場合がある。すなわち、図5中のステップS515の文字部の強調処理の方法によっては、記載されている文字をユーザが識別できなくなるなど、適切な強調表示がなされない場合がある。例えば強調処理において、対応するグリフ二値画像の画素値が0である画素はそのままにし、グリフ二値画像の画素値が1である画素に対しては各画素値(r, g, b)を予め決められた色(例えば(0, 0, 0))に各々変化させるようなビューアがあったとする。この場合、本実施形態のFont01は四角で塗りつぶすグリフを使用しているので、強調表示すると図17のようなページ画像が表示されることになってしまう。図17のような状態になってしまうと、文字画像が見えなくなってしまう、ユーザの視認性が悪くなってしまふ。

30

【 0 1 3 1 】

そこで、上述したように本実施形態では、このようなビューア・アプリケーションで利用される場合を想定し、ステップS208で電子文書を生成する際に、異なるグリフを有するフォントデータを複数種類、格納するようにしている。したがって、ユーザは再度図3のステップS301のフォントID指定の処理を用いて、別のフォントIDを指定すれば強調表示の方法を変更できる。

40

【 0 1 3 2 】

S301でデフォルトのフォントIDに“Font02”を指定し、強調表示をおこなった場合のページ画像表示の例を図14に示す。図6中の記述612に指定されるグリフ二値画像を用いて各画素値(r, g, b)を(0, 0, 0)へと変化させた強調表示がそれぞれ描画されると、図14のようなページ画像が生成される。つまり、“Font02”で用いるグリフは、描画矩形単位内下部の1024x64の領域を塗潰す水平方向直線状のグリフであるから、この部分の画素値を(0, 0, 0)にした場合、図14のように文字に下線が付与されたかのように強調表示されることになる。したがって、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できるとともに、文

50

字画像の視認性も確保できる。

【0133】

ここで図3に戻る。ステップS308では、ユーザが検索・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選択する。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S309に進む。

【0134】

ステップS309では、 $n = n + k$ とし、ステップS305に戻って以降の処理を繰り返す。

【0135】

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するとともに、異なる字形を有するフォントを複数格納する。この電子文書に対して、ユーザは強調表示時に用いるデフォルトの字形を選択、すなわち字形の切替指示をできるようにする。したがって、各ビューア・アプリケーション或いはその文書画像に最適な字形に切り替えて強調表示させることができる。検索キーワードに一致する箇所が見やすい形で強調表示されたページ表示を、ユーザが確認しながら検索を進めていくことが可能である。

【0136】

本実施形態の電子文書は、単純な字形からなるフォントデータを一つの文字に対して複数、内部で持ち、文書内の透明文字の描画する際に、上記単純な字形の1つを選択して描画できるようになる。また、各フォントデータにおいて、1つの字形を複数の文字種に対して共通して利用するようにしている。したがって、使用されるフォントデータを電子文書内部に保持するが、字形データは少なく済むので、当該電子文書のファイルサイズ(データ容量)は小さく抑えることができる。更に、いくつかのフォントデータを格納しているため、検索の強調表示時には視認性や操作性のよい表示が可能となる。

【0137】

<実施形態2>

次に、本発明の第2の実施形態(実施形態2)について図面を用いて説明する。

【0138】

図15は、本実施形態2により生成された電子文書の例である。前述の実施形態1と同様に、画像処理装置100が電子文書を生成・送信し、画像処理装置110が受信・閲覧・検索するものとする。

【0139】

図15中の1501および1513は電子文書の開始、終了を表す記述である。1502および1506は1ページ目の描画の開始、終了を表わす記述である。1503は1ページ目の画像データ描画の記述である。1504および1505は1ページ目の文字描画の記述である。また、1507および1510は2ページ目の描画の開始、終了を表わす記述である。1508は2ページ目の画像データ描画の記述である。1509は2ページ目の文字描画の記述である。1511および1512はこの電子文書で用いられるフォントデータの記述である。

【0140】

実施形態1では、図3ステップS301において、デフォルトのフォントIDをユーザが選択していたが、本実施形態2では、閲覧処理を行うアプリケーション(ビューア)がその選択のための判断を行えるように、電子文書を生成する。その場合、図4のステップS408のフォントデータ記述において、要素にアプリケーションが判断する為の属性を追加することになる。本実施形態では、図15中の要素1511および1512ではShapeという属性により、フォントデータの形状(フォントデータの特徴)が容易に判別できるようにしている。この場合、このフォントデータのShape属性を判断基準として用いることにより、アプリケーションはそのアプリケーションで強調表示を行うときに適した表示用のフォントIDを選択できるようになる。なお、上記追加属性は一例であり、フォントデータの特徴以外に、閲覧処理をするアプリケーシ

10

20

30

40

50

ョンの名前や種類などを属性として記述してもよい。この場合、これを判断基準として、例えば、アプリケーションが自らのアプリケーション名が含まれているフォントデータの判断を行い、デフォルト表示用のフォントIDを決定する。

【0141】

本実施形態2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し、異なる字形を有するフォントを複数記述し、アプリケーションがどのフォントを使用すべきか判断する為の属性を持つ。この電子文書に対しては、アプリケーション（ビューアなど）が上記属性を判断して字形を自動的に選択して当該アプリケーションで強調表示するのに適した字形に切り換えて表示することができる。また、検索キーワードに一致する箇所が強調表示され、且つ文字画像の視認性が自動的に確保されたページ表示をユーザが確認しながら検索を進めていくことが可能である。

10

【0142】

上述のように、本実施形態2における電子文書は、文書内で記述したすべての透明文字の描画にあたって、アプリケーション（ビューアなど）が属性を判断して複数の字形から一つの字形を選択して描画するように記述されている。また、各フォントデータにおいて、一つの字形を複数の文字種に対して共通して利用するようにしている。そのため、電子文書が当該電子文書内で使用されるフォントデータを保持するが、字形データは少なく済むので、電子文書のファイルサイズ（データ容量）は小さく抑えることができる。更に、いくつかのフォントデータを格納しているため、検索の強調表示時にはアプリケーションに最適な表示が可能である。

20

【0143】

<実施形態3>

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。

【0144】

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。

30

【0145】

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。

【0146】

<実施形態4>

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。

40

【0147】

以上、本発明の諸実施形態について説明した。

50

【 0 1 4 8 】

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。

【 0 1 4 9 】

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、不揮発性のメモリカード、ROMなどを用いることができる。

【 0 1 5 0 】

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。

【 0 1 5 1 】

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 】

【 0 1 5 2 】

【 図 1 】 本発明の実施形態 1 および 2 の画像処理装置の構成例を表すブロック図である。

【 図 2 】 本発明の実施形態 1 および 2 の電子文書生成処理の例を表すフローチャートである。

【 図 3 】 本発明の実施形態 1 および 2 の電子文書検索・閲覧処理の例を表すフローチャートである。

【 図 4 】 図 2 中ステップ S 2 0 8 にて行われる電子文書データ生成処理の例を表すフローチャートである。

【 図 5 】 図 3 中ステップ S 3 0 7 にて行われるページの描画処理の例を表すフローチャートである。

【 図 6 】 本発明の実施形態 1 により生成される電子文書の一例である。

【 図 7 】 本発明の実施形態 1 および 2 にて処理されるページ画像の一例である。

【 図 8 】 本発明の実施形態 1 および 2 にて行われる領域解析処理例を表す図である。

【 図 9 】 本発明の実施形態 1 および 2 にて生成される領域データの一例である。

【 図 1 0 】 本発明の実施形態 1 および 2 にて行われる文字認識処理例を表す図である。

【 図 1 1 】 本発明の実施形態 1 および 2 にて生成される文字コード列データの例である。

【 図 1 2 】 本発明の実施形態 1 および 2 にて利用される文字コード配列テーブルの一例である。

【 図 1 3 】 本発明の実施形態 1 および 2 で実施される検索箇所が強調されたページ表示例である。

【 図 1 4 】 本発明の実施形態 1 で実施される検索箇所が強調されたページ表示例である。

【 図 1 5 】 本発明の実施形態 2 により生成される電子文書の一例である。

【 図 1 6 】 本発明の実施形態 1 にて表示されるUIの一例である。

【 図 1 7 】 本発明の実施形態 1 で実施される検索箇所が強調されたページ表示例である。

10

20

30

40

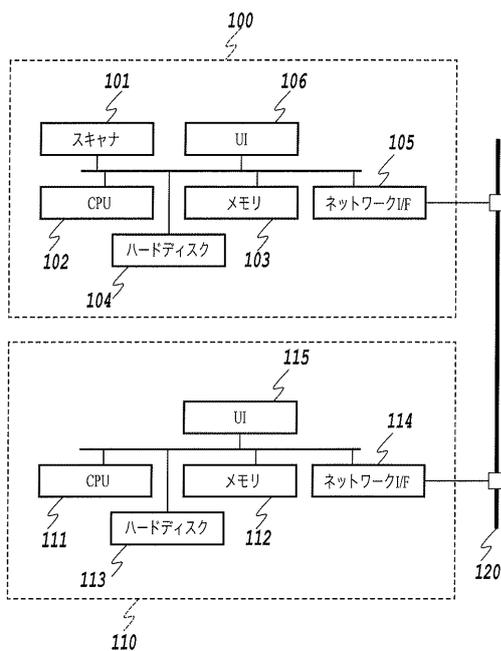
50

【符号の説明】

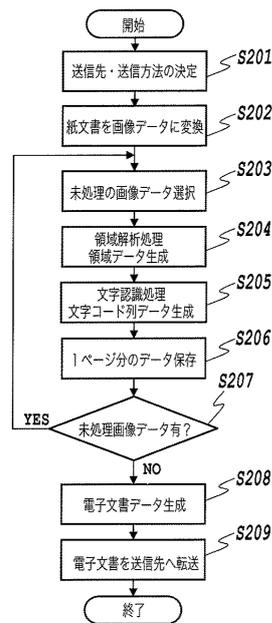
【0153】

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

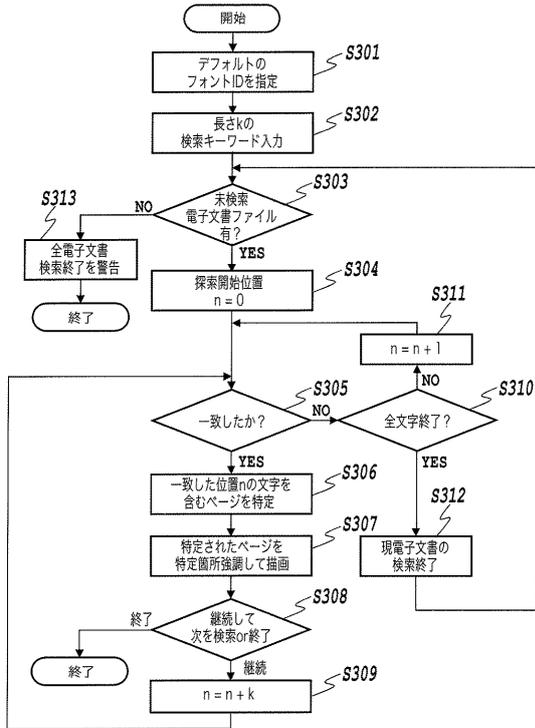
【図1】



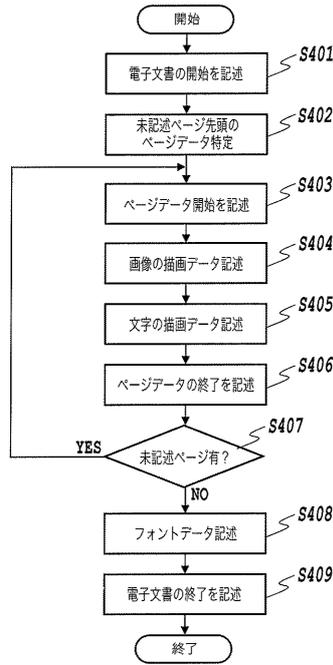
【図2】



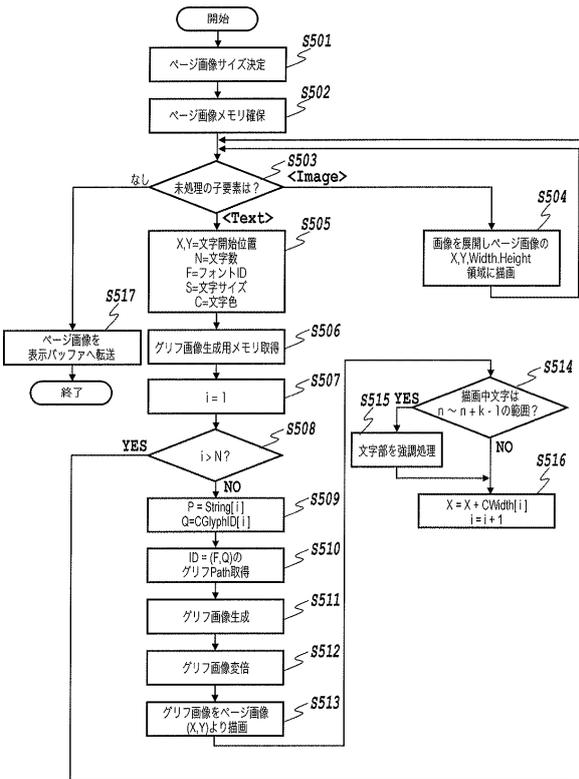
【図3】



【図4】



【図5】

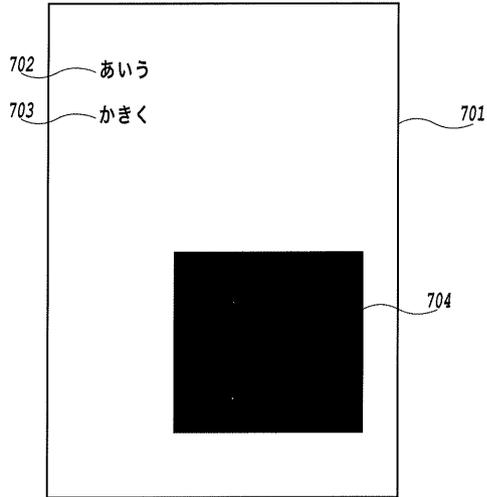


【図6】

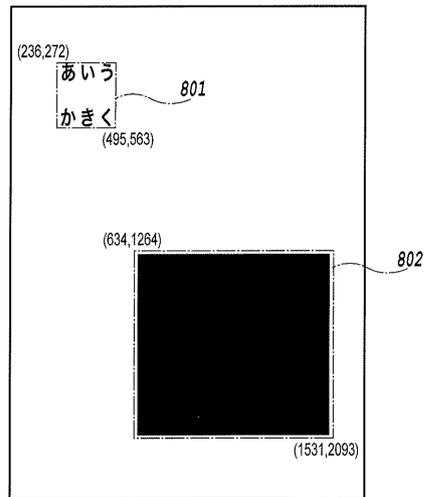
```

601 { <?xml version="1.0"?>
602 { <Document>
603 { <Page Width="1680" Height="2376" Dpi="200" Number="1">
604 { <Image X="0" Y="0" Width="1680" Height="2376"
605 { Data="ACAf49sw89q34rs6dbc23cfasx...a3B52zAD"/>
606 { <Text X="236" Y="272" Size="97" Direction="Horizontal"
607 { Color="0,0,0,255" String="0x2422,0x2424,0x2426"
608 { CWidth="104,96,59", CGlyphId="0,0,0"/>
609 { <Text X="236" Y="472" Size="92" Direction="Horizontal"
610 { Color="0,0,0,255" String="0x242b,0x242d,0x242f"
611 { CWidth="113,101,47", CGlyphId="0,0,0"/>
612 { </Page>
613 { </Document>
  
```

【図7】



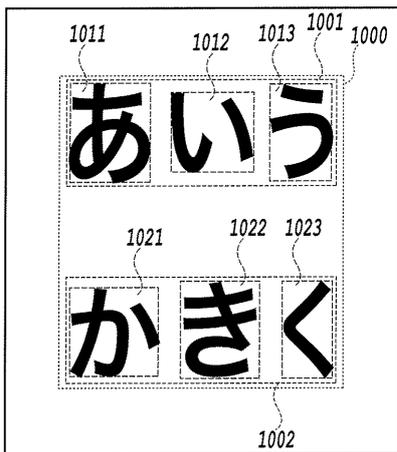
【図8】



【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	コード
1	236	272	80	97	0x2422
2	339	248	79	75	0x2424
3	434	274	59	95	0x2426

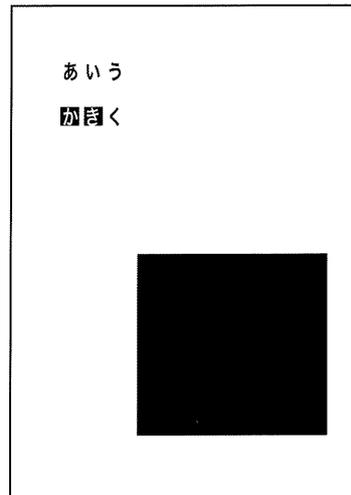
1101

文字	x	y	width	height	コード
1	236	474	87	85	0x242b
2	348	472	75	92	0x242d
3	448	472	47	89	0x242f

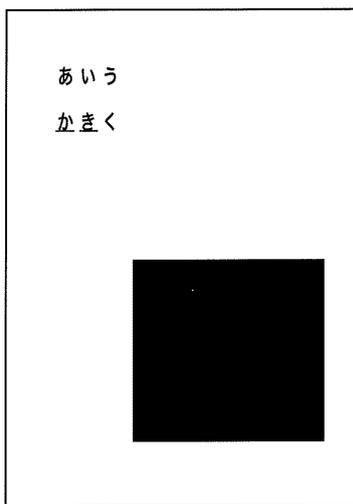
【 図 1 2 】

文字	コード	コード
0	0x2422	1093
1	0x2424	1100
2	0x2426	1107
3	0x242b	1250
4	0x242d	1257
5	0x242f	1264
6	0x242b	1601
7	0x242d	1608
8	0x242f	1615

【 図 1 3 】



【 図 1 4 】



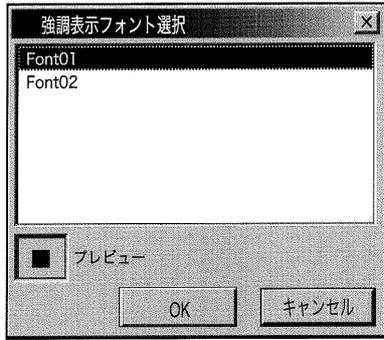
【 図 1 5 】

```

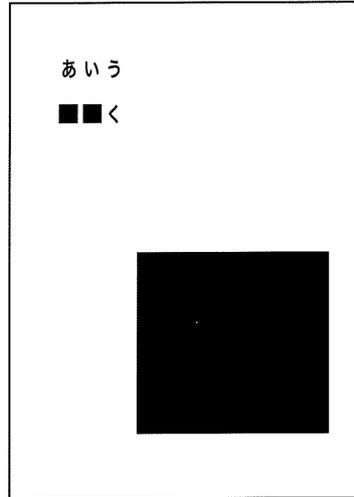
1501 { <?xml version="1.0"?>
1502 { <Document>
1503 { <Image X="0" Y="0" Width="1680" Height="2376"
      Data="ACAf49sw89q34rs6dbc23cfasx...a3E52zAD"/>
1504 { <Text X="236" Y="272" Size="97" Direction="Horizontal"
      Color="0,0,0,255" String="0x2422,0x2424,0x2426"
      CWidth="104,96,59", CGlyphId="0,0,0"/>
1505 { <Text X="236" Y="472" Size="92" Direction="Horizontal"
      Color="0,0,0,255" String="0x242b,0x242d,0x242f"
      CWidth="113,101,47", CGlyphId="0,0,0"/>
1506 { </Page>
1507 { <Page Width="1680" Height="2376" Dpi="200" Number="2">
1508 { <Image X="0" Y="0" Width="1680" Height="2376"
      Data="sa80AN2qx7sa23Basd2x...aAqw8D2dDJ"/>
1509 { <Text X="136" Y="872" Size="92" Direction="Horizontal"
      Color="0,0,0,255" String="0x242b,0x242d,0x242f"
      CWidth="113,101,47", CGlyphId="0,0,0"/>
1510 { </Page>
1511 { <Font Id="Font01" Shape="BOX">
      <Glyph Id="0" Path="M0,0 V-1024 H1024 V1024 f"/>
      </Font>
1512 { <Font Id="Font02" Shape="UNDER_LINE">
      <Glyph Id="0" Path="M0,0 V-64 H1024 V64 f"/>
      </Font>
1513 { </Document>

```

【図 16】



【図 17】



フロントページの続き

(72)発明者 中西 恵子
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 成瀬 博之

(56)参考文献 特開2005-259017(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F 17/20 - 17/26

G06T 1/00