

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6512763号
(P6512763)

(45) 発行日 令和1年5月15日 (2019.5.15)

(24) 登録日 平成31年4月19日 (2019.4.19)

(51) Int.Cl.

F I

G 0 6 T 11/00 (2006.01)

G 0 6 T 11/00

G 0 6 F 3/12 (2006.01)

G 0 6 F 3/12 3 4 5

請求項の数 12 (全 17 頁)

(21) 出願番号 特願2014-146086 (P2014-146086)
 (22) 出願日 平成26年7月16日 (2014.7.16)
 (65) 公開番号 特開2016-24495 (P2016-24495A)
 (43) 公開日 平成28年2月8日 (2016.2.8)
 審査請求日 平成29年7月13日 (2017.7.13)

(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 110001243
 特許業務法人 谷・阿部特許事務所
 (72) 発明者 大石 哲
 東京都大田区下丸子3丁目30番2号 キ
 ヤノン株式会社内

審査官 岡本 俊威

(56) 参考文献 特開2009-101662 (JP, A)

最終頁に続く

(54) 【発明の名称】 情報処理装置、情報処理方法、プログラム

(57) 【特許請求の範囲】

【請求項1】

編集されたドキュメントから、元ドキュメントに対する編集内容を特定する編集内容特定手段と、

ドキュメントを作成するアプリケーション毎のオブジェクトの表現形式に関する特徴を示した特徴情報を取得する取得手段と、

前記取得手段で取得した特徴情報に基づいて、前記元ドキュメントに含まれるオブジェクトの表現形式を特定する表現形式特定手段と、

前記編集内容特定手段で特定した編集内容の表現形式と、前記表現形式特定手段で特定したオブジェクトの表現形式とが合うように、前記編集内容の表現形式を変換する変換手段と、

前記変換手段で変換された表現形式の編集内容を、前記元ドキュメントに統合する統合手段と、

を有することを特徴とする情報処理装置。

【請求項2】

編集されたドキュメントから、元ドキュメントに対する編集内容を特定する編集内容特定手段と、

前記元ドキュメントに含まれるオブジェクトの描画コマンドの種別とその個数とに関する情報を含むリストを作成することにより、前記元ドキュメントに含まれるオブジェクトの表現形式を特定する表現形式特定手段と、

10

20

前記編集内容特定手段で特定した編集内容の表現形式と、前記表現形式特定手段で特定したオブジェクトの表現形式とが合うように、前記編集内容の表現形式を変換する変換手段と、

前記変換手段で変換された表現形式の編集内容を、前記元ドキュメントに統合する統合手段と、

を有することを特徴とする情報処理装置。

【請求項 3】

編集されたドキュメントから、元ドキュメントに対する編集内容を特定する編集内容特定手段と、

前記元ドキュメントに含まれるオブジェクトの表現形式を特定する表現形式特定手段と、

前記編集内容特定手段で特定した編集内容の表現形式と、前記表現形式特定手段で特定したオブジェクトの表現形式とが合うように、前記編集内容の表現形式を変換する変換手段と、

前記変換手段で変換された表現形式の編集内容を、前記元ドキュメントに統合する統合手段と、

を有し、

前記変換手段は、オブジェクトの色が異なること、オブジェクトのフォントサイズが異なること、オブジェクトに連続性がないことのうち少なくとも 1 つの条件を満たす場合、前記変換を行わない、

ことを特徴とする情報処理装置。

【請求項 4】

前記変換手段は、前記編集内容特定手段で特定した編集内容の表現形式と、前記表現形式特定手段で特定したオブジェクトの表現形式とが合うように、前記編集内容に含まれる属性と描画コマンドとのうちの少なくともいずれかを変換することを特徴とする請求項 1 乃至 3 のいずれか 1 項に記載の情報処理装置。

【請求項 5】

前記特徴情報は、アプリケーションの名称と当該アプリケーションにおけるオブジェクトの表現形式の特徴とを対応付けた情報であり、

前記表現形式特定手段は、前記元ドキュメントの作成に使用したアプリケーションの名称と一致する名称のアプリケーションが前記特徴情報に含まれる場合に、当該一致するアプリケーションに対応するオブジェクトの表現形式を、前記元ドキュメントにおけるオブジェクトの表現形式として特定する

ことを特徴とする請求項 1 に記載の情報処理装置。

【請求項 6】

前記変換手段は、前記編集内容を変換する候補が複数ある場合、当該複数の候補のうち、前記編集内容に係る描画コマンドを前記リストにおいて個数が 1 以上である描画コマンドの表現形式に変換することを特徴とする請求項 2 に記載の情報処理装置。

【請求項 7】

前記変換手段は、前記リスト内に、前記個数が 1 以上の描画コマンドが複数ある場合、前記編集内容に係る描画コマンドを個数が最多の描画コマンドに変換することを特徴とする請求項 6 に記載の情報処理装置。

【請求項 8】

前記編集内容特定手段は、前記元ドキュメントと、前記編集されたドキュメントとを比較することによって、前記編集内容を特定することを特徴とする請求項 1 乃至 7 のいずれか 1 項に記載の情報処理装置。

【請求項 9】

情報処理装置の編集内容特定手段が、編集されたドキュメントから、元ドキュメントに対する編集内容を特定するステップと、

前記情報処理装置の取得手段が、ドキュメントを作成するアプリケーション毎のオブジ

10

20

30

40

50

エクトの表現形式に関する特徴を示した特徴情報を取得するステップと、

前記情報処理装置の表現形式特定手段が、前記取得するステップで取得した特徴情報に基づいて、前記元ドキュメントに含まれるオブジェクトの表現形式を特定するステップと

、

前記情報処理装置の変換手段が、前記特定された編集内容の表現形式が、前記特定されたオブジェクトの表現形式と合うように、前記編集内容の表現形式を変換するステップと

、

前記情報処理装置の統合手段が、前記変換された表現形式の編集内容を、前記元ドキュメントに統合するステップと、

を含むことを特徴とする情報処理方法。

10

【請求項 10】

情報処理装置の編集内容特定手段が、編集されたドキュメントから、元ドキュメントに対する編集内容を特定する第 1 の特定ステップと、

前記情報処理装置の表現形式特定手段が、前記元ドキュメントに含まれるオブジェクトの描画コマンドの種別とその個数とに関する情報を含むリストを作成することにより、前記元ドキュメントに含まれるオブジェクトの表現形式を特定する第 2 の特定ステップと、

前記情報処理装置の変換手段が、前記第 1 の特定ステップで特定した編集内容の表現形式と、前記第 2 の特定ステップで特定したオブジェクトの表現形式とが合うように、前記編集内容の表現形式を変換するステップと、

前記情報処理装置の統合手段が、前記変換するステップで変換された表現形式の編集内容を、前記元ドキュメントに統合するステップと、

を含むことを特徴とする情報処理方法。

20

【請求項 11】

情報処理装置の編集内容特定手段が、編集されたドキュメントから、元ドキュメントに対する編集内容を特定する第 1 の特定ステップと、

前記情報処理装置の表現形式特定手段が、前記元ドキュメントに含まれるオブジェクトの表現形式を特定する第 2 の特定ステップと、

前記情報処理装置の変換手段が、前記第 1 の特定ステップで特定した編集内容の表現形式と、前記第 2 の特定ステップで特定したオブジェクトの表現形式とが合うように、前記編集内容の表現形式を変換するステップと、

前記情報処理装置の統合手段が、前記変換するステップで変換された表現形式の編集内容を、前記元ドキュメントに統合するステップと、

を含み、

前記変換手段は、オブジェクトの色が異なること、オブジェクトのフォントサイズが異なること、オブジェクトに連続性がないことのうち少なくとも 1 つの条件を満たす場合、前記変換を行わない、

30

ことを特徴とする情報処理方法。

【請求項 12】

コンピュータを、請求項 1 乃至 8 のいずれか 1 項に記載の情報処理装置の各手段として機能させるためのプログラム。

40

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、情報処理装置、情報処理方法、プログラムに関する。

【背景技術】

【0002】

元ドキュメント（例えば、PDF のドキュメント）に対し、情報（例えば、「Confidential」という文字列からなるアノテーション）を追加するといった編集を行うことがよくある。特許文献 1 には、PDF で作成された元ドキュメントを他のフォーマットに変換した上で編集を行なった後、編集箇所を PDF コマンドに再度変換してその編集部分のみ PDF の元ド

50

キュメントにマージする技術が開示されている。

【先行技術文献】

【特許文献】

【0003】

【特許文献1】特開2005 - 10841号公報

【発明の概要】

【発明が解決しようとする課題】

【0004】

上記のような情報を追加する編集を行う際、どのように情報を追加するのが好ましいかは、ケースバイケースである。例えば、「Confidential」という文字列からなるアノテーションを新たに追加する場合には、テキストとして追加するのが好ましい場合もあれば、グラフィックとして追加するのが好ましい場合もあれば、イメージとして追加するのが好ましい場合もある。

10

【0005】

例えば、元ドキュメントに含まれる文字列がテキストで表されている場合であれば、新たに追加された文字列も同様にテキストとして追加されるべきである。一方、元ドキュメントに含まれる文字列がグラフィックで表されている場合であれば、新たな追加された文字列も同様にグラフィックとして追加されるべきである。さらに、元ドキュメントに含まれる文字列がイメージで表されている場合であれば、新たな追加された文字列も同様にイメージとして追加されるべきである。そうしないと、編集後のドキュメント（即ち、情報を追加した後のドキュメント）を印刷した際に同種類のオブジェクト間で色味等が異なって違和感が生じてしまうからである。これは、一般に印刷装置では、オブジェクトの属性に応じた画像処理を行うためである。例えば、テキストの印刷時にはテキスト専用の画像処理（例えば、エッジをシャープにする処理）、イメージの印刷時には写真専用の画像処理（例えば、ぼかし処理）がかけられる一方で、グラフィックの印刷時にはそれら画像処理はなされない。

20

【0006】

よって、文字列をテキストで表している元ドキュメントに対して新たに文字列をイメージで追加して印刷すると、元ドキュメント内に存在していた文字列と、新たに追加された文字列とでは見た目が印刷結果において異なり、違和感が生じることになる。

30

【課題を解決するための手段】

【0007】

本発明に係る情報処理装置は、編集されたドキュメントから、元ドキュメントに対する編集内容を特定する編集内容特定手段と、ドキュメントを作成するアプリケーション毎のオブジェクトの表現形式に関する特徴を示した特徴情報を取得する取得手段と、前記取得手段で取得した特徴情報に基づいて、前記元ドキュメントに含まれるオブジェクトの表現形式を特定する表現形式特定手段と、前記編集内容特定手段で特定した編集内容の表現形式と、前記表現形式特定手段で特定したオブジェクトの表現形式とが合うように、前記編集内容の表現形式を変換する変換手段と、前記変換手段で変換された表現形式の編集内容を、前記元ドキュメントに統合する統合手段と、を有することを特徴とする。

40

【発明の効果】

【0008】

本発明によれば、元ドキュメントに対し編集を行なった場合においても、その印刷時に色味の違いといった違和感を生じさせることがない。

【図面の簡単な説明】

【0009】

【図1】実施例1に係るドキュメント編集システムの構成の一例を示す図である。

【図2】サーバのハードウェア構成の一例を示すブロック図である。

【図3】サーバのソフトウェア構成の一例を示すブロック図である。

【図4】差分抽出の様子を説明する図である。

50

【図 5】コマンドリスト作成処理の流れを示すフローチャートである。

【図 6】コマンドリストの一例を示す図である。

【図 7】描画コマンド属性決定処理の流れを示すフローチャートである。

【図 8】アプリ特徴情報の一例を示す図である。

【図 9】アプリ特徴情報に基づいて編集箇所の描画コマンドが変換される一例を示す図である。

【図 10】コマンドリストに基づいて編集箇所の描画コマンドが変換される一例を示す図である。

【図 11】編集箇所の描画コマンド（変換後）が元ドキュメントに統合される様子を説明する図である。

10

【図 12】編集箇所の描画コマンドがそのまま維持される場合の一例を説明する図である。

【図 13】フォーマット変換を伴う編集において適用した場合の一例を説明する図である。

【発明を実施するための形態】

【0010】

以下、添付図面を参照して、本発明を好適な実施例に従って詳細に説明する。なお、以下の実施例において示す構成は一例にすぎず、本発明は図示された構成に限定されるものではない。

【実施例 1】

20

【0011】

図 1 は、本実施例に係るドキュメント編集システムの構成の一例を示す図である。ドキュメント編集システム 100 は、PC 101、携帯端末 102、印刷装置 103、Web アプリケーションサーバ（以下、サーバ）104 で構成される。

【0012】

PC 101 はクライアント端末としてのコンピュータ、携帯端末 102 は携帯電話やタブレット端末等である。PC 101 や携帯端末 102 では、サーバ 104 から所定のフォーマットの（例えば、PDF）のドキュメントを受け取り、編集用アプリケーションを使ってドキュメントを編集することができる。印刷装置 103 は、例えばインクジェット方式や電子写真方式のプリンタであり、編集後のドキュメントの印刷に用いられる。PC 101、携

30

帯端末 102、印刷装置 103 は、ネットワーク 106 を介して互いに接続されており、ネットワーク 106 はインターネット 107 に繋がっている。

【0013】

サーバ 104 は、1 台のサーバから構成されてもよいし、複数台のサーバから構成されてもよい。

【0014】

なお、ドキュメント編集システム 100 は、アプリケーションサービスとしてのドキュメント編集アプリケーションを提供するサーバ（群）を別途備えていてもよい。

【0015】

図 2 は、サーバ 104 のハードウェア構成の一例を示すブロック図である。

40

【0016】

サーバ 104 は、CPU 201、RAM 202、ROM 203、HDD 204 およびネットワーク I/F 205、内部バス 206 で構成されている。

【0017】

ROM 203 或いは HDD 204 には、オペレーティングシステムやアプリケーションなどの制御プログラムが格納されている。CPU 201 は、当該制御プログラムを必要に応じて ROM 203 或いは HDD 204 から RAM 202 上へ読み出して実行することで、コンピュータとしての機能を発揮する。

【0018】

ネットワーク I/F 205 は、内部バス 206 とインターネット 106 とを繋ぐインタ

50

フェースであり、インターネット106を介して接続されている他の装置との間でPDF形式のドキュメント等の各種データの送受信を行う。

【0019】

図3は、サーバ104のソフトウェア構成の一例を示すブロック図である。以下に示すソフトウェアモジュールとしての各部は、上述したCPU201がROM203等に格納されたプログラムをRAM202に展開し実行することにより実現される。ただし、これら各部の一部がハードウェアロジックで構成されていてもよい。

【0020】

ドキュメント管理部300は、編集処理の対象となる所定のフォーマットで作成されたドキュメントのデータを管理する。具体的には、以下のような処理を行なう。

10

【0021】

- ・データベース（不図示のDBサーバ等、外部に設けてもよい）に格納されている既存のドキュメント（以下、元ドキュメント）のデータをユーザ指示に応じて取得
- ・編集作業を行なうPC101や携帯端末102に元ドキュメントのデータを送信
- ・編集が施されたドキュメント（以下、編集ドキュメント）のデータをPC101や携帯端末102から受信

なお、ドキュメントのフォーマットとして本実施例ではPDFを例として説明を行なうものとするが、当然のことながら他のフォーマットでも構わない。

【0022】

編集反映処理部310は、受信した編集ドキュメントに係る編集内容を、元ドキュメントに反映させる処理を行なう。元ドキュメントへの編集内容の反映には、以下の4つの処理が含まれる。

20

【0023】

- 1) 編集ドキュメントからの編集箇所の抽出（編集内容の特定）
- 2) 元ドキュメント及び編集箇所の解析
- 3) 抽出された編集箇所の表現形式（描画コマンドの属性）の決定
- 4) 表現形式が決定した編集箇所（描画コマンド）の元ドキュメントへの統合

【0024】

編集箇所抽出部311は、上述の1)に記載した、編集ドキュメントから編集箇所を抽出して編集内容を特定する。具体的には、編集ドキュメントと元ドキュメントとの差分（編集箇所）を、既知の方法であるテキスト差分抽出法や画像差分抽出法によって抽出する。図4は、差分抽出の様子を説明する図である。401は編集がなされる前の元ドキュメントであり、「あいう」の文字列と三角形の図形が描画されている。402は、元ドキュメント401における文字列と図形を描画するためのPDF描画コマンドを示している。403は、編集がなされた後の編集ドキュメントであり、元から存在した「あいう」の文字列と三角形の図形に、「かきく」の文字列が新たに追加されているのが分かる。404は、編集ドキュメント403における文字列と図形を描画するためのPDF描画コマンドを示している。405は、元ドキュメント401と編集ドキュメント403との差分（編集箇所）であり、上述の「かきく」の文字列が編集箇所として抽出されることを示している。406は、抽出された編集箇所に対応するPDF描画コマンドである。抽出された編集箇所の情報は、RAM202に格納される。

30

40

【0025】

ドキュメント解析部312は、上述の2)に記載した、元ドキュメント及び編集箇所を解析する処理を行なう。元ドキュメントの解析においては、当該元ドキュメントで描画されるすべてのオブジェクトの描画コマンドの種別を特定した上で、描画コマンド別の個数情報を含むリスト（以下、コマンドリスト）が作成される。編集箇所の解析においては、編集箇所の描画コマンドの種別が特定される。元ドキュメント及び編集箇所の解析処理の詳細については後述する。

【0026】

描画コマンド属性決定部313は、上述の3)に記載した、編集箇所の描画コマンドの

50

属性を決定する処理を行なう。具体的には、編集箇所抽出部 3 1 1 で抽出された編集箇所の描画コマンドが元ドキュメントに合ったものとなるよう、当該編集箇所の表現形式（描画コマンドの属性）を適宜変換する処理を行なう。この描画コマンドの属性を決定する処理の詳細についても後述する。

【 0 0 2 7 】

ドキュメント統合部 3 1 4 は、上述の 4) に記載した、表現形式が決定した編集箇所を元ドキュメントへ統合する処理を行なう。具体的には、描画コマンド属性決定部 3 1 3 で決定した編集箇所の描画コマンドを、元ドキュメントの描画コマンドに統合する。

【 0 0 2 8 】

通信部 3 2 0 は、ドキュメント管理部 3 0 0 からの指示に従い、元ドキュメントや編集ドキュメントといった各種データの送受信を行う。例えば、編集内容が反映された元ドキュメントのデータが通信部 3 2 0 を介して印刷装置 1 0 3 に送られる。

【 0 0 2 9 】

印刷装置 1 0 3 では、受け取ったドキュメントの描画コマンドを解釈して 2 値画像に変換し、当該 2 値画像に基づく画像をプリントエンジンを用いて記録媒体上に形成する。その際、冒頭で述べたとおり、オブジェクトの属性に応じた画像処理がなされる。例えば、テキストの印刷時にはテキスト専用の画像処理（例えば、エッジをシャープにする処理）、イメージの印刷時には写真専用の画像処理（例えば、ぼかし処理）、グラフィックの印刷時にはいずれの画像処理も行わないといった制御がなされる。

【 0 0 3 0 】

本実施例では、編集によって新たに追加された情報に対応する描画コマンドは、元ドキュメントで用いられている描画コマンドに合わせて変換された上で、元ドキュメントに統合されるので、属性の齟齬が生じない。つまり、編集内容が反映されたドキュメントの印刷処理においては、属性が統一されて同じ画像処理が適用されることになる。したがって、編集内容が反映されたドキュメントを印刷しても、新たに追加されたオブジェクトと元からあるオブジェクトとの間で色味が異なるといった問題が生じない。

【 0 0 3 1 】

次に、ドキュメント解析部 3 1 2 における処理の詳細について説明する。ここでは、元ドキュメントの解析処理（コマンドリスト作成処理）の場合を例に説明するものとする。図 5 は、コマンドリスト作成処理の流れを示すフローチャートである。

【 0 0 3 2 】

ステップ 5 0 1 において、ドキュメント解析部 3 1 2 は、元ドキュメント内の任意の描画コマンドが、Text コマンドかどうかを判定する。描画コマンドの種別の判定は、予め描画コマンドを “ /Type/ ” 以下に続く内容で分類しておくことで識別可能である。例えば、“ /Type/Text ” で始まる描画コマンドはText コマンド、“ /Type/Image ” で始まる描画コマンドはImage コマンド、“ /Type/Graphics ” で始まる描画コマンドはGraphics コマンドといった具合に分類しておけばよい。判定の結果、Text コマンドであれば、ステップ 5 0 2 に進む。一方、Text コマンドでなければ、ステップ 5 0 5 に進む。

【 0 0 3 3 】

ステップ 5 0 2 において、ドキュメント解析部 3 1 2 は、Text コマンドと判定された描画コマンドがPathで構成されているかどうかを判定する。Textを描画する方法は、点列でアウトラインを描画するやり方と、イメージで描画するやり方とに大別できる。アウトラインを構成する点列をPathといい、Text コマンドがこのPathで構成されている場合は、ステップ 5 0 3 に進む。一方、Text コマンドがPathで構成されていない場合は、ステップ 5 0 4 に進む。

ステップ 5 0 3 において、ドキュメント解析部 3 1 2 は、描画コマンドがPathで構成されるText コマンドであることを示す「PathText」のカウンタをカウントアップする。

【 0 0 3 4 】

ステップ 5 0 4 において、ドキュメント解析部 3 1 2 は、描画コマンドがPathで構成されない（Imageで構成される）Text コマンドであることを示す「ImageText」のカウンタを

10

20

30

40

50

カウントアップする。

【0035】

ステップ505において、ドキュメント解析部312は、描画コマンドがImageコマンドかどうかを判定する。判定の結果、Imageコマンドであれば、ステップ506に進む。一方、Imageコマンドでなければ、ステップ511に進む。

【0036】

ステップ506において、ドキュメント解析部312は、描画コマンドが全面イメージを指定するImageコマンドかどうかを判定する。具体的には、ページの領域とイメージの領域とを比較し、両者の差分が所定の範囲内（例えば5%以内）である場合に、当該Imageコマンドは全面イメージを指定する描画コマンドであると判定する。判定の結果、全面イメージのImageコマンドであれば、ステップ507に進む。一方、全面イメージのImageコマンドでなければ、ステップ508に進む。

10

【0037】

ステップ507において、ドキュメント解析部312は、描画コマンドが全面イメージのImageコマンドであることを示す「ScanImage」のカウンタをカウントアップする。

【0038】

ステップ508において、ドキュメント解析部312は、描画コマンドがグラデーションのイメージを指定するImageコマンドかどうかを判定する。グラデーションイメージかどうかは、当該Imageコマンドで描画されるイメージの内容が、LinearGradiation, Radial Gradiation, 3点Gradiation等の所定のパターンに一致するかどうかによって（パターンマッチング）判定することができる。判定の結果、グラデーションイメージのImageコマンドであれば、ステップ509に進む。一方、グラデーションイメージのImageコマンドでなければ、ステップ510に進む。

20

【0039】

ステップ509において、ドキュメント解析部312は、描画コマンドがグラデーションイメージのImageコマンドであることを示す「GradationImage」のカウンタをカウントアップする。

【0040】

ステップ510において、ドキュメント解析部312は、描画コマンドがその他のイメージのImageコマンドであることを示す「Image」のカウンタをカウントアップする。

30

【0041】

ステップ511において、ドキュメント解析部312は、描画コマンドがGraphicsコマンドかどうかを判定する。判定の結果、Graphicsコマンドであれば、ステップ512に進む。一方、Graphicsコマンドでなければ、ステップ517に進む。

【0042】

ステップ512において、ドキュメント解析部312は、描画コマンドが、単位面積当たりのPath数が所定の閾値（例えば100個）以上のGraphicsコマンドであるかどうかを判定する。判定の結果、単位面積当たりのPath数が所定の閾値以上のGraphicsコマンドであれば、ステップ513に進む。一方、単位面積当たりのPath数が所定の閾値以上のGraphicsコマンドでなければ、ステップ514に進む。

40

【0043】

ステップ513において、ドキュメント解析部312は、描画コマンドが、多数のPathで構成される文字を表すGraphicsコマンドであることを示す「GraphicsText」のカウンタをカウントアップする。

【0044】

ステップ514において、ドキュメント解析部312は、描画コマンドがグラデーションのグラフィックを指定するGraphicsコマンドかどうかを判定する。グラデーションのグラフィックかどうかは、例えば、描画コマンド内に“/ShadingType/”があるかどうかで判定することができる。判定の結果、グラデーションのグラフィックを指定するGraphicsコマンドであれば、ステップ515に進む。一方、グラデーションのグラフィックを指定

50

するGraphicsコマンドでなければ、ステップ5 1 6に進む。

【0 0 4 5】

ステップ5 1 5において、ドキュメント解析部3 1 2は、描画コマンドがグラデーションのグラフィックを指定するGraphicsコマンドであることを示す「GradationGraphics」のカウンタをカウントアップする。

【0 0 4 6】

ステップ5 1 6において、ドキュメント解析部3 1 2は、描画コマンドがグラデーション以外のグラフィックを指定するGraphicsコマンドであることを示す「Graphics」のカウンタをカウントアップする。

【0 0 4 7】

ステップ5 1 7において、ドキュメント解析部3 1 2は、描画コマンドがその他の描画を指定することを示す「Other」のカウンタをカウントアップする。

【0 0 4 8】

ステップ5 1 8において、ドキュメント解析部3 1 2は、処理対象の元ドキュメントについて未処理の描画コマンドがあるかどうかを判定する。未処理の描画コマンドがあればステップ5 0 1に戻り、次の描画コマンドについての処理を続行する。一方、未処理の描画コマンドがなければ、ステップ5 1 9に進む。

【0 0 4 9】

ステップ5 1 9において、ドキュメント解析部3 1 2は、全てのカウンタのカウント結果（カウント値）を集計して、元ドキュメントに含まれる描画コマンドの種別とその個数の情報からなるコマンドリストを生成する。図6は、本ステップで生成されるコマンドリストの一例を示している。6 0 1は、前述の元ドキュメント4 0 1に対応するPDF描画コマンド4 0 2と同じものであり、これを解析した結果、6 0 2のようなコマンドリストが得られる。コマンドリスト6 0 2から、元ドキュメント4 0 1が、3つのGraphicsTextコマンドと1つのGraphicsコマンドという計4つの描画コマンドで構成されていることが分かる。なお、コマンドリスト6 0 2には、アプリケーションの名称「アプリケーションB」が、元ドキュメントの作成に使用されたアプリケーションを特定するための情報として付されている。

【0 0 5 0】

以上が、コマンドリスト作成処理の内容であり、生成されたコマンドリストは、RAM 2 0 2等に格納される。なお、編集箇所の解析処理も基本的には同じである。ただし、編集箇所の解析処理では、描画コマンドの種別毎の個数情報は不要なのでカウント処理は行わず、編集箇所の描画コマンドの種別がすべて判定された段階で処理は完了する。

【0 0 5 1】

続いて、描画コマンド属性決定部3 1 3における、編集箇所に係る描画コマンドの属性を決定する処理の詳細について説明する。図7は、描画コマンド属性決定処理の流れを示すフローチャートである。

【0 0 5 2】

ステップ7 0 1において、描画コマンド属性決定部3 1 3は、元ドキュメント及び編集箇所の解析結果（すなわち、元ドキュメントのコマンドリスト、及び編集箇所の描画コマンドのデータ）をRAM 2 0 2等から取得する。

【0 0 5 3】

ステップ7 0 2において、描画コマンド属性決定部3 1 3は、特定のフォーマット（本実施例ではPDF）のドキュメントを作成可能なアプリケーションにおける表現形式の特徴を示した情報（以下、アプリ特徴情報）を取得する。ここで、アプリ特徴情報について説明する。

【0 0 5 4】

同じフォーマットかつ同じ内容の画像を含んだドキュメントであっても、それぞれの作成に使用したアプリケーションが異なれば、その表現の仕方も異なり得る。例えば、PDFのドキュメントを作る際、グラデーションを1つのイメージ（ビットマップデータ）とし

10

20

30

40

50

て表現するアプリケーションもあれば、多数のグラフィックス（即ち、図形データ）の集合体として表現するアプリケーションもある。また、文字を文字コードで表現するものもあれば、グラフィックス（即ち、図形データ）で表現するものもある。あるいは矩形を4頂点の座標位置が指定されたグラフィックスとして表現するものもあれば、イメージとして表現するものもある。こういったアプリケーション毎のオブジェクトの表現形式に関する特徴の情報を集めてリスト形式にまとめたものを、本明細書では「アプリ特徴情報」と呼んでいる。そして、このようなアプリ特徴情報を用いることで、各アプリケーションの特徴に沿って、編集箇所に係る描画コマンドを必要に応じて変換することが可能となる。図8は、アプリ特徴情報の一例を示す図である。図8のアプリ特徴情報には、PDFのドキュメントを作成可能な4種類のアプリケーションA~Dについての特徴が記載されている。アプリケーションAの特徴「Gradation by Image」はグラデーションをイメージで描画するという特徴を表している。アプリケーションBの特徴「Text by Graphics」は、テキストをグラフィックで描画するという特徴を表している。アプリケーションCの特徴「Rectangle by Image」は、矩形をイメージで描画するという特徴を表している。アプリケーションDの特徴「Rectangle by Fill」は、矩形を4頂点の座標位置が指定されたグラフィックスで描画するという特徴を表している。このように、アプリ特徴情報は、各アプリケーションについて、アプリケーションの名称と当該アプリケーションにおけるオブジェクトの表現形式の特徴とを対応付けた情報となっている。

10

【0055】

ステップ703において、描画コマンド属性決定部313は、コマンドリストを参照し、編集の対象となった元ドキュメントの作成に使用したアプリケーションと一致するアプリケーションがアプリ特徴情報に存在するかどうかを判定する。元ドキュメントの作成に使用したアプリケーションと一致するアプリケーションがアプリ特徴情報に存在する場合は、ステップ704に進む。一方、元ドキュメントの作成に使用したアプリケーションと一致するアプリケーションがアプリ特徴情報に存在しない場合は、ステップ706に進む。

20

【0056】

ステップ704において、描画コマンド属性決定部313は、該当するアプリケーションの特徴情報を参照して、編集箇所の描画コマンドが、当該アプリケーションの特徴に係る描画コマンドに該当するかどうかを判定する。例えば、元ドキュメントがアプリケーションB（図8を参照）で作成されている場合、その特徴は「Text by Graphics」であるので、編集箇所の描画コマンドがTextコマンドであれば「該当」と判定されることになる。判定の結果、編集箇所の描画コマンドが、元ドキュメントの作成アプリケーションの特徴に係る描画コマンドに該当する場合は、ステップ705に進む。一方、編集箇所の描画コマンドが、元ドキュメントの作成アプリケーションの特徴に係る描画コマンドに該当しない場合は、ステップ706に進む。

30

【0057】

ステップ705において、描画コマンド属性決定部313は、アプリ特徴情報に基づき、必要に応じて、元ドキュメントの特徴に沿うように編集箇所の描画コマンドを変換する。図9は、アプリ特徴情報に基づいて編集箇所の描画コマンドが変換される一例を示す図である。図9において、枠901は、編集ドキュメント403における文字列と図形を描画するためのPDF描画コマンド401における編集箇所の描画コマンド（追加された文字列「かきく」に対応）を示している。枠902は、変換後の編集箇所の描画コマンドを示しており、元ドキュメントを作成したアプリケーションBの特徴である「Text by Graphics」に従い、TextコマンドからGraphicsコマンド（GraphicsText）に変換されている。この変換は、不図示のRIPモジュールで行われる。なお、「かきく」に対応する編集箇所の描画コマンドが当初からGraphicsコマンドであった場合には、そのまま維持されることになる。

40

【0058】

ステップ706において、描画コマンド属性決定部313は、編集箇所の描画コマンド

50

に対し、複数の候補が存在するかどうかを判定する。前述の図 6 で示したコマンドリストから明らかなように、例えばテキストの場合、PathTextなのか、ImageTextなのか、GraphicsTextなのかといった複数の候補が存在する。同様に、例えばグラデーションの場合、GradationImageなのか、GradationGraphicsなのかといった複数の候補が存在する。このように編集箇所の描画コマンドについて複数の候補が存在するかどうか判定される。判定の結果、複数の候補が存在する場合はステップ 707 に進む。一方、複数の候補が存在しない場合は、変換の必要がないので編集箇所の描画コマンドに手を加えることなくステップ 708 に進む。

【0059】

ステップ 707 において、描画コマンド属性決定部 313 は、元ドキュメントのコマンドリストに基づき、必要に応じて編集箇所の描画コマンドを変換する。具体的には、複数の候補の内、コマンドリストにおいてカウント数が 1 以上である描画コマンドに変換（コマンドリスト内にカウント数が 1 以上の描画コマンドが複数ある場合はカウント数が最多のものに変換）する。図 10 は、コマンドリストに基づいて編集箇所の描画コマンドが変換される一例を示す図である。図 10 において、1001 は「A」から「Z」までの文字列からなる PDF の元ドキュメント、1002 はその描画コマンド（GraphicsText で構成）を示している。1003 は描画コマンド 1002 から作成されたコマンドリストであり、GraphicsText のカウント数のみが 26 でその他のカウント数はすべて 0 である。そして、1004 は「123」の文字列が追加された編集ドキュメント、1005 は文字列「123」に対応する編集箇所の描画コマンド（PathText で構成）、1006 はコマンドリスト 1003 に従って変換された編集箇所の描画コマンド（GraphicsText で構成）を示している。編集箇所の描画コマンド 1005 は“PathText”であったが、コマンドリスト 1003 でカウント数が唯一 1 以上である“GraphicsText”と一致するように変換されているのが分かる。

【0060】

ステップ 708 において、描画コマンド属性決定部 313 は、未処理の編集箇所の描画コマンドがあるかどうかを判定する。未処理の編集箇所の描画コマンドがあれば、ステップ 703 に戻って次の編集箇所の描画コマンドについて処理を続行する。一方、未処理の編集箇所の描画コマンドがなければ、本処理を終える。

【0061】

以上が、描画コマンド属性決定処理の内容である。この処理によって必要に応じて変換がなされた編集箇所の描画コマンドのデータはドキュメント統合部 314 に送られ、元ドキュメントに統合される。図 11 は、編集箇所の描画コマンド（変換後）が元ドキュメントに統合される様子を説明する図である。図 11 において、1101 は元ドキュメントであり図 6 で示した元ドキュメント 601 と同じものである。1102 は統合後の元ドキュメントであり、アプリ特徴情報に従って“Text”から“Graphics（GraphicsText）”へと変換された編集箇所の描画コマンド 1103（図 9 で示した変換後の描画コマンド 902 と同じもの）が組み込まれている。

【0062】

このように、本実施例では、元ドキュメントに編集内容を反映させる際に、編集箇所の描画コマンドが元ドキュメントの特徴に沿ったものと変換される。

【0063】

なお、本実施例においては、編集後のドキュメントデータそのものを受け取り、サーバ 104 内の編集箇所抽出部 311 で編集箇所を抽出していた。しかし、このような態様に限定されるものではない。例えば、ドキュメントの編集を行なう PC 101 や携帯端末 102 から編集箇所を抽出し、抽出された編集箇所の情報（編集箇所の描画コマンドのデータ）のみをサーバ 104 に送るようにしてもよい。これにより、サーバ側での処理負担を軽減することができる。

【0064】

また、PC 101 や携帯端末 102 から編集内容（位置・大きさ・色を指定して「Confid

10

20

30

40

50

ential」のアノテーションを追加する旨の指示等)の情報だけをサーバ104に送るようにしてもよい。この場合、サーバ104において、元ドキュメントに沿った描画コマンドとなるよう、取得した情報で特定される編集内容の描画コマンドを決定した上で元ドキュメントに反映すればよい。

【0065】

さらには、本実施例においてWebサービスのサーバ104で行った処理を、クラウドコンピューティングでのアプリケーションによって実現してもよい。或いは、本システムで行なったすべての処理(印刷処理を除く)を、サーバ104、PC101、モバイル端末102といった情報処理装置の単体で実現するようにしてもよい。

【0066】

本実施例によれば、元ドキュメントに追加したい情報に対応する描画コマンドは、元ドキュメントで用いられている描画コマンドに合わせて変換された上で、元ドキュメントに統合される。よって、統合後のドキュメントにおいて属性の不一致が生じることがなく、同種類のオブジェクトには同じ画像処理が適用されることになる。したがって、統合後のドキュメントを印刷しても、オブジェクト間で色味が異なるといった問題が生じるのを防ぐことができる。

【実施例2】

【0067】

実施例1では、編集箇所の描画コマンドの属性が元ドキュメントに沿ったものとなるよう描画コマンドを変換していた。次に、編集箇所の描画コマンドを一定条件下で変換しないようにする態様について、実施例2として説明する。

【0068】

具体的には、前述の図7のフローチャートのステップ705やステップ707における処理において、色、フォントサイズ、連続性といった特徴を考慮して、変換するかどうかを決定する。図12は、編集箇所の描画コマンドがそのまま維持される場合の一例を説明する図である。図12において、1201は編集の対象となった元ドキュメントであり、「あいう」の文字列がGraphicsコマンドで描画されている。1202は、編集がなされた後の編集ドキュメントであり、「あいう」の文字列に、1回り大きなサイズの「かきく」の文字列が新たにTextコマンドで追加されている。1203は、該編集内容が反映された元ドキュメントであり、一定の条件下では新たに追加された文字列「かきく」がGraphicsコマンドに変換されることなくTextコマンドのままで反映されることを示している。

【0069】

実施例1であれば、追加された文字列「かきく」は、元ドキュメントの特徴に沿ってGraphicsコマンドに変換された上で追加されることになる。本実施例では、追加される文字の色やフォントサイズが元ドキュメントに存在する文字の色やフォントサイズと一致するかどうかを判定し、一致しない場合は、異なる意図の描画と判断してデフォルト(ここではTextコマンド)のままで追加する。

【0070】

この場合において、例えば追加される文字のフォントサイズが元ドキュメントに存在する文字のフォントサイズと一致するかどうかの判定は、所定の変化率(例えば30%)の範囲内かどうかによって決定すればよい。色についても同様である。したがって、上記1回り大きなサイズの「かきく」の文字列が、所定の変化率の範囲内であればGraphicsコマンドに変換され、所定の変化率の範囲外であればTextコマンドのままで維持されることになる。

【0071】

また、元ドキュメントにおける文字列「あいう」と追加された文字列「かきく」の距離によって文字列としての連続性があるかどうかを判定し、連続性がないと判定された場合は、違う意図の描画としてデフォルトのままで追加するようにしてもよい。例えば、追加された文字列の所定の範囲(例えば1000pixel内)にある文字列の描画コマンドのみ変換時に参照することでこれを実現することが可能である。

【実施例 3】

【0072】

実施例 1 及び 2 では、同一フォーマット間での編集を前提にしていた。上記実施例で述べた内容は、フォーマット変換を伴う編集時にも適用可能である。

【0073】

図 1 3 は、フォーマット変換を伴う編集時に適用した場合の一例を説明する図である。ここでは、図 1 3 (a) に示す様に、まず PDF から SVG に変換し、SVG において編集を行ない、その後再度 PDF に変換する場合を想定する。

【0074】

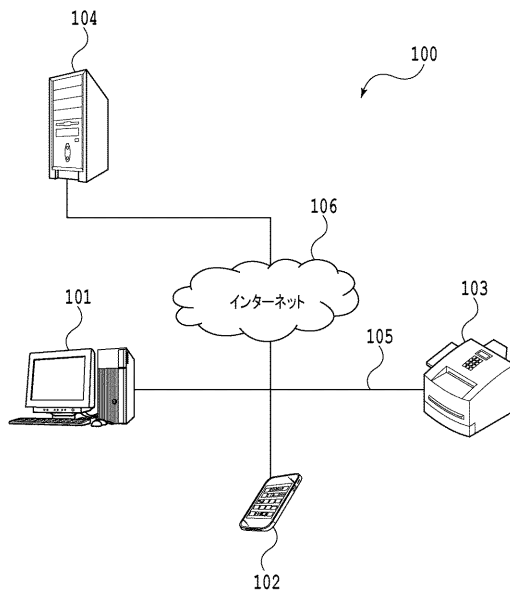
PDF に再度変換する時に、全ての SVG コマンドを PDF コマンドに変換すると、元の PDF ドキュメントとは異なる構成の PDF ドキュメントになってしまう。そのため、編集部分の SVG コマンドのみ PDF コマンドに変換して元の PDF ドキュメントに統合する (図 1 3 (b))。この SVG コマンドから PDF コマンドへの変換の際に、実施例 1 及び 2 の手法を適用することで、元の PDF ドキュメントの特徴を踏まえたコマンド体系の、編集後の PDF ドキュメントを得ることができる。

【0075】

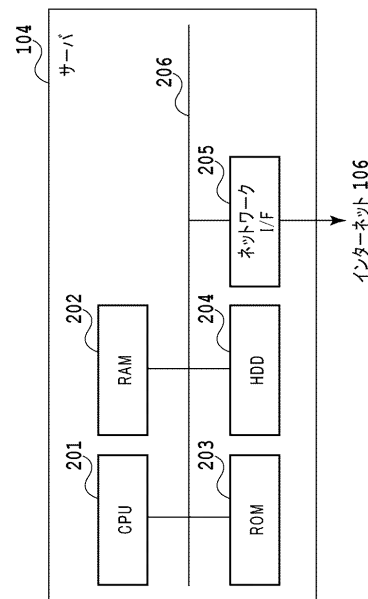
(その他の実施例)

また、本発明は、以下の処理を実行することによっても実現される。即ち、上述した実施形態の機能を実現するソフトウェア (プログラム) を、ネットワーク又は各種記憶媒体を介してシステム或いは装置に供給し、そのシステム或いは装置のコンピュータ (または CPU や MPU 等) がプログラムを読み出して実行する処理である。

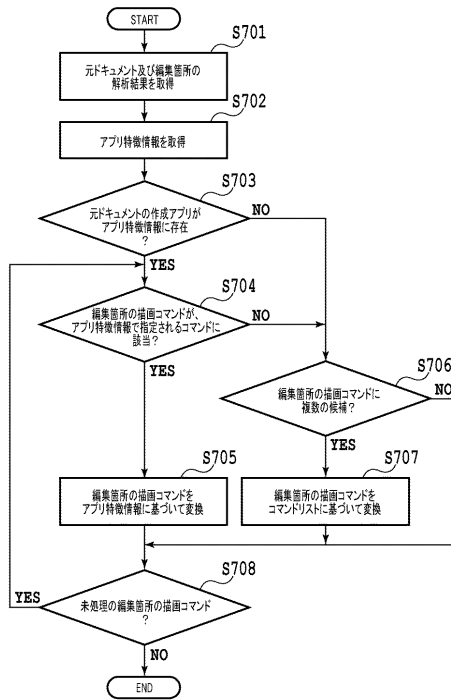
【図 1】



【図 2】



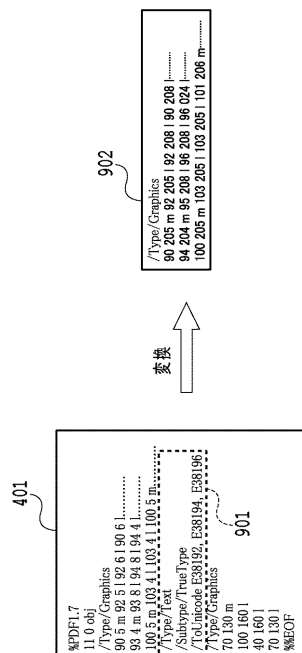
【 図 7 】



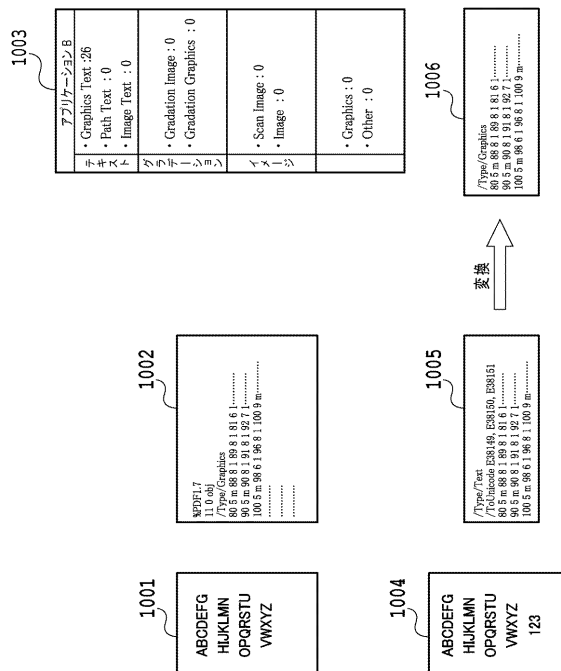
【 図 8 】

アプリケーション A : Gradation by Image
アプリケーション B : Text by Graphics
アプリケーション C : Rectangle by Image
アプリケーション D : Rectangle by Fill

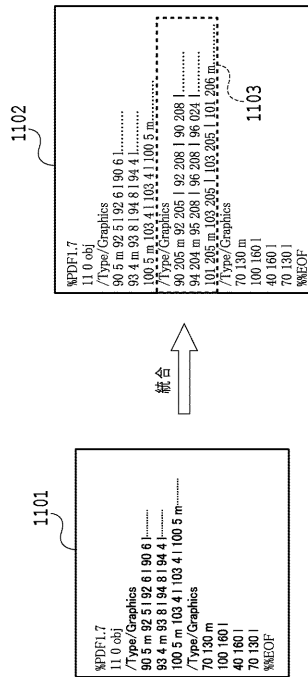
【 図 9 】



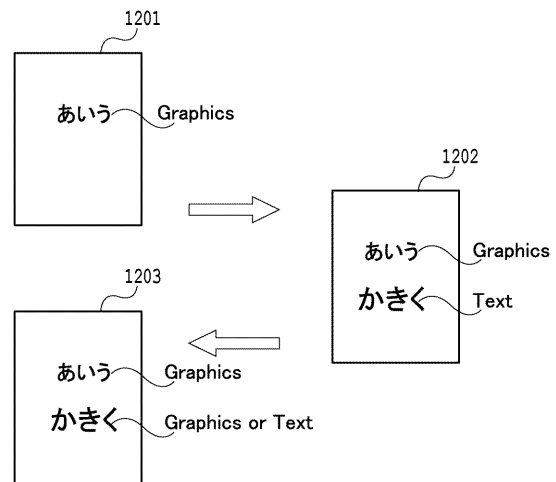
【 図 1 0 】



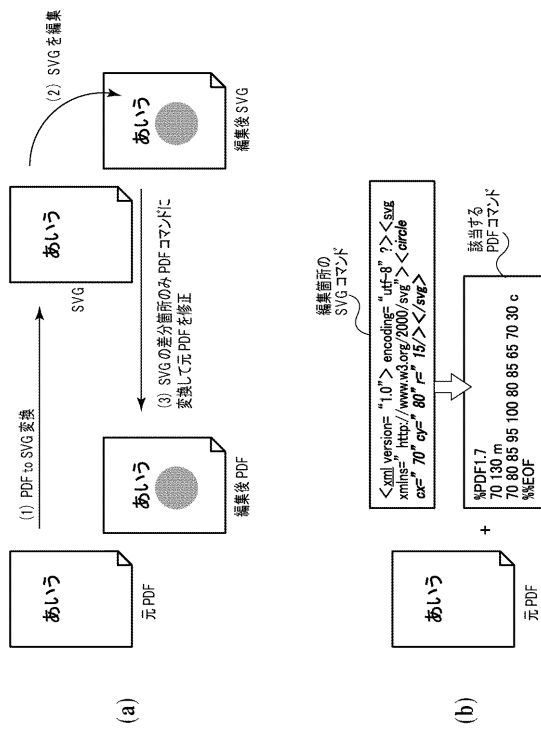
【 図 1 1 】



【 図 1 2 】



【 図 1 3 】



フロントページの続き

(58)調査した分野(Int.Cl. , D B 名)

G 0 6 T 1 1 / 0 0 - 1 1 / 8 0

G 0 6 F 3 / 1 2