

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5339667号  
(P5339667)

(45) 発行日 平成25年11月13日(2013.11.13)

(24) 登録日 平成25年8月16日(2013.8.16)

(51) Int.Cl.

F I

G09G 5/36 (2006.01)  
 G09G 5/22 (2006.01)  
 G06F 17/21 (2006.01)  
 G06T 11/80 (2006.01)  
 G06T 11/00 (2006.01)

G09G 5/36 530C  
 G09G 5/36 520P  
 G09G 5/36 530Y  
 G09G 5/22 670  
 G06F 17/21 501T

請求項の数 16 (全 21 頁) 最終頁に続く

(21) 出願番号 特願2006-100389 (P2006-100389)  
 (22) 出願日 平成18年3月31日(2006.3.31)  
 (65) 公開番号 特開2007-17945 (P2007-17945A)  
 (43) 公開日 平成19年1月25日(2007.1.25)  
 審査請求日 平成21年3月26日(2009.3.26)  
 (31) 優先権主張番号 特願2005-171657 (P2005-171657)  
 (32) 優先日 平成17年6月10日(2005.6.10)  
 (33) 優先権主張国 日本国(JP)

前置審査

(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 100076428  
 弁理士 大塚 康徳  
 (74) 代理人 100112508  
 弁理士 高柳 司郎  
 (74) 代理人 100115071  
 弁理士 大塚 康弘  
 (74) 代理人 100116894  
 弁理士 木村 秀二  
 (74) 代理人 100130409  
 弁理士 下山 治  
 (74) 代理人 100134175  
 弁理士 永川 行光

最終頁に続く

(54) 【発明の名称】 情報処理装置及びその方法

(57) 【特許請求の範囲】

【請求項1】

ベクタ画像を記述した構造化文書进行处理する情報処理装置であって、  
 前記構造化文書を解析して得られた前記ベクタ画像を表示部に表示させる表示手段と、  
 前記表示部においてユーザにより指定された領域を検出する検出手段と、  
 前記ベクタ画像に含まれる文字オブジェクトの一部が前記指定された領域に含まれるか  
 否かを判定する判定手段と、

前記判定手段によって前記指定された領域に含まれると判定された前記文字オブジェクトと前記指定された領域の外縁との交差点の座標を、前記文字オブジェクトの種類および前記文字オブジェクトの前記ベクタ画像の座標に基づいて決定する決定手段と、

前記決定手段によって決定された前記交差点の座標に基づき前記文字オブジェクトの記述を変更することにより、(i)前記判定手段によって前記指定された領域に含まれないと判定された文字オブジェクトの記述が削除され前記指定された領域内のベクタ画像を記述し且つ(ii)前記指定された領域内に含まれると判定された文字オブジェクトの少なくとも一部を曲線で近似することによって得られる文字オブジェクトを記述した構造化文書を作成する作成手段と、

を有することを特徴とする情報処理装置。

【請求項2】

前記作成手段は、前記指定された領域の位置を基準として前記指定された領域内のベクタ画像に含まれる文字オブジェクトの位置を記述した前記構造化文書を作成することを特

10

20

徴とする請求項 1 に記載の情報処理装置。

【請求項 3】

前記作成手段は、前記判定手段によって部分的に前記指定された領域内に含まれると判定された文字オブジェクトの記述を、前記指定された領域内に含まれる部分の文字オブジェクトの記述に変更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 4】

前記作成手段は、前記判定手段によって部分的に前記指定された領域内に含まれると判定された文字オブジェクトが、直線的な形状で表現される文字オブジェクトの場合、前記文字オブジェクトの位置情報および前記指定された領域の位置情報に基づいて、前記指定された領域内に含まれる部分の文字オブジェクトの記述に変更することを特徴とする請求項 3 に記載の情報処理装置。

10

【請求項 5】

前記作成手段は、前記判定手段によって部分的に前記指定された領域内に含まれると判定された文字オブジェクトが、円、楕円または曲線を含む文字オブジェクトである場合、前記文字オブジェクトの位置情報および前記指定された領域の位置情報に基づいて、前記指定された領域内に含まれる部分の文字オブジェクトをベジェ曲線で近似することによって得られる文字オブジェクトの記述に変更することを特徴とする請求項 3 に記載の情報処理装置。

【請求項 6】

前記作成手段は、前記文字オブジェクトの記述の変更後のデータ量が変更前のデータ量よりも大きくなる場合には、前記文字オブジェクトの記述の変更を禁止することを特徴とする請求項 3 に記載の情報処理装置。

20

【請求項 7】

前記検出手段によって検出された領域内のベクタ画像を拡大する拡大手段を更に有し、  
前記作成手段は、前記拡大手段によって拡大された状態のベクタ画像に含まれる文字オブジェクトを記述した構造化文書を作成することを特徴とする請求項 1 乃至 6 のいずれか 1 項に記載の情報処理装置。

【請求項 8】

ベクタ画像を記述した構造化文書进行处理する情報処理装置の情報処理方法であって、  
前記構造化文書を解析して得られた前記ベクタ画像を表示部に表示させる表示ステップと、

30

前記表示部においてユーザにより指定された領域を検出する検出ステップと、

前記ベクタ画像に含まれる文字オブジェクトの一部が前記指定された領域に含まれるかを判定する判定ステップと、

前記判定ステップにおいて前記指定された領域に含まれると判定された前記文字オブジェクトと前記指定された領域の外縁との交差点の座標を、前記文字オブジェクトの種類および前記文字オブジェクトの前記ベクタ画像の座標に基づいて決定する決定ステップと、

前記決定ステップにおいて決定された前記交差点の座標に基づき前記文字オブジェクトの記述を変更することにより、( i ) 前記判定ステップにおいて前記指定された領域に含まれないと判定された文字オブジェクトの記述が削除され前記指定された領域内のベクタ画像を記述し且つ ( i i ) 前記指定された領域内に含まれると判定された文字オブジェクトの少なくとも一部を曲線で近似することによって得られる文字オブジェクトを記述した構造化文書を作成する作成ステップと、

40

を有することを特徴とする情報処理装置の情報処理方法。

【請求項 9】

前記作成ステップにおいて、前記指定された領域の位置を基準として前記指定された領域内のベクタ画像に含まれる文字オブジェクトの位置を記述した前記構造化文書を作成することを特徴とする請求項 8 に記載の情報処理装置の情報処理方法。

【請求項 10】

前記作成ステップにおいて、部分的に前記指定された領域内に含まれると判定された文

50

字オブジェクトの記述を、前記指定された領域内に含まれる部分の文字オブジェクトの記述に変更することを特徴とする請求項 9 に記載の情報処理装置の情報処理方法。

【請求項 1 1】

前記作成ステップにおいて、部分的に前記指定された領域内に含まれると判定された文字オブジェクトが、直線的な形状で表現される文字オブジェクトの場合、前記文字オブジェクトの位置情報および前記指定された領域の位置情報に基づいて、前記指定された領域内に含まれる部分の文字オブジェクトの記述に変更することを特徴とする請求項 1 0 に記載の情報処理装置の情報処理方法。

【請求項 1 2】

前記作成ステップにおいて、部分的に前記指定された領域内に含まれると判定された文字オブジェクトが、円、楕円または曲線を含む文字オブジェクトである場合、前記文字オブジェクトの位置情報および前記指定された領域の位置情報に基づいて、前記指定された領域内に含まれる部分の文字オブジェクトをベジェ曲線で近似することによって得られる文字オブジェクトの記述に変更することを特徴とする請求項 1 0 に記載の情報処理装置の情報処理方法。

【請求項 1 3】

前記作成ステップは、前記文字オブジェクトの記述の変更後のデータ量が変更前のデータ量よりも大きくなる場合には、前記文字オブジェクトの記述の変更を禁止することを特徴とする請求項 1 0 に記載の情報処理装置の情報処理方法。

【請求項 1 4】

前記検出ステップで検出された領域内のベクタ画像を拡大する拡大ステップを更に有し、前記作成ステップは、前記拡大ステップで拡大された状態のベクタ画像に含まれる文字オブジェクトを記述した構造化文書を作成することを特徴とする請求項 8 乃至 1 3 のいずれか 1 項に記載の情報処理装置の情報処理方法。

【請求項 1 5】

請求項 8 乃至 1 4 のいずれか 1 項に記載の情報処理装置の情報処理方法を実行するためのプログラム。

【請求項 1 6】

請求項 1 5 に記載のプログラムを記憶したコンピュータにより読み書き可能な記憶媒体。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

本発明は、構造化文書で記述された複数のオブジェクトからなるベクタ画像を処理可能な情報処理装置及びその方法に関するものである。

【背景技術】

【0 0 0 2】

ベクタ形式の画像は、解像度によらず高品位な表示が得られる等のメリットがある。このようなベクタ形式の画像を作成、編集するグラフィックス描画アプリケーションソフトとして、例えばAdobe Systems社のイラストレータ（登録商標）が挙げられる。このAdobe Systems社のイラストレータ（登録商標）は、その指定領域内のオブジェクトを移動させて編集することが可能である。例えば、ベクタ画像を高速に描画する技術として、特許文献 1 がある。

【特許文献 1】特開 2 0 0 4 - 1 6 4 2 0 8 号公報

【発明の開示】

【発明が解決しようとする課題】

【0 0 0 3】

X M L ベースでベクタ画像を表現するための言語として、S V G が存在する。例えば、図 1 に示すベクタ画像を記述した S V G ファイルを、処理能力の低い機器に通信ネットワークを介して転送した場合を想定する。この場合、本来送信する必要のあるグラフィック

10

20

30

40

50

領域が領域 1 0 0 3 だとしても、図 1 のオブジェクト 1 0 0 0 ~ 1 0 0 2 のすべてをグラフィック化しなければならず機器に負担がかかってしまう。また、機器のメモリの容量もその分必要となり、転送時間もかかってしまう。特に図 2 乃至図 4 のように、指定領域 2 0 0 0 , 3 0 0 0 , 4 0 0 1 に含まれる図形や文字のオブジェクトの部分が少ない場合には、不要なデータで多くの記憶容量を消費する場合が考えられる。

【 0 0 0 4 】

本発明は、上記従来技術の欠点を解決することにある。

【 0 0 0 5 】

また本願発明の特徴は、複数のオブジェクトを含むベクタ画像において、指定領域に含まれるグラフィックスデータだけをベクトルデータとして抽出して保存する情報処理装置及びその方法を提供することにある。

【課題を解決するための手段】

【 0 0 0 6 】

上記目的を達成するために本発明の一態様に係る情報処理装置は以下のような構成を備える。即ち、ベクタ画像を記述した構造化文書进行处理する情報処理装置であって、

前記構造化文書を解析して得られた前記ベクタ画像を表示部に表示させる表示手段と、前記表示部においてユーザにより指定された領域を検出する検出手段と、前記ベクタ画像に含まれる文字オブジェクトの一部が前記指定された領域に含まれるか否かを判定する判定手段と、前記判定手段によって前記指定された領域に含まれると判定された前記文字オブジェクトと前記指定された領域の外縁との交差点の座標を、前記文字オブジェクトの種類および前記文字オブジェクトの前記ベクタ画像の座標に基づいて決定する決定手段と、前記決定手段によって決定された前記交差点の座標に基づき前記文字オブジェクトの記述を変更することにより、( i ) 前記判定手段によって前記指定された領域に含まれないと判定された文字オブジェクトの記述が削除され前記指定された領域内のベクタ画像を記述し且つ ( i i ) 前記指定された領域内に含まれると判定された文字オブジェクトの少なくとも一部を曲線で近似することによって得られる文字オブジェクトを記述した構造化文書を作成する作成手段と、を有することを特徴とする。

【 0 0 0 8 】

尚、この課題を解決するための手段は、本願発明の特徴の全てを列挙しているものではなく、特許請求の範囲に記載された他の請求項及びそれら特徴群の組み合わせも発明になり得る。

【発明の効果】

【 0 0 0 9 】

本発明によれば、複数のオブジェクトを含むベクタ画像において、指定領域に含まれるベクタ画像を抽出し、抽出したベクタ画像に対応する構造化文書を作成して保存できるという効果がある。その結果、機器の処理の負担が軽減される。

【発明を実施するための最良の形態】

【 0 0 1 0 】

以下、添付図面を参照して本発明の好適な実施の形態を詳しく説明する。尚、以下の実施の形態は特許請求の範囲に係る本発明を限定するものでなく、また本実施の形態で説明されている特徴の組み合わせの全てが本発明の解決手段に必須のものとは限らない。

【 0 0 1 1 】

図 5 は、本発明の実施の形態に係るシステムの概要を示した構成図である。

【 0 0 1 2 】

図 5 において、スキャナ 1 0 1 は L A N 1 0 2 に接続されている。L A N 1 0 2 には、クライアントとしての P C 1 0 3、ネットワークプリンタ 1 0 4、ファイルサーバ 1 0 5 が接続されている。

【 0 0 1 3 】

図 6 は、本実施の形態に係る情報処理装置 ( P C ) 1 0 3 の構成を示すブロック図である。

10

20

30

40

50

## 【 0 0 1 4 】

図 6 において、CPU 2 0 1 はシステム制御部であり、この情報処理装置 1 0 3 の全体を制御している。ROM 2 0 2 は、CPU 2 0 1 の制御プログラムや各種固定データを格納している。RAM 2 0 3 は、SRAM や DRAM 等で構成され、CPU 2 0 1 による制御処理の実行時にプログラム制御変数等を格納する。また各種設定パラメータ、各種ワーク用バッファも、この RAM 2 0 3 に格納される。記憶部 2 0 4 はハードディスク (HD) 等で構成され、文書や画像等のスキャン画像を格納する。操作パネル 2 0 6 は、キーボード、タッチパネル等で構成され、オペレータが各種入力操作を行うのに使用される。表示部 2 0 7 は、液晶等で構成され、オペレータに提示するメッセージやメニューコマンド等を表示する。LAN i / f 2 0 8 は、LAN 2 0 9 に接続するためのインターフェースである。USB i / f 2 1 0 は、USB 回線 2 1 1 に接続するためのインターフェースである。

10

## 【 0 0 1 5 】

以上の構成において、ユーザは、操作パネル 2 0 6 を用いて、表示部 2 0 7 に表示されているベクタ画像画像内の一部分の範囲を領域指定する。そして、その指定領域の位置に基づいて得られたベクタ画像を、LAN 1 0 2 を介してプリンタ 1 0 4、ファイルサーバ 1 0 5 に転送することができる。

## 【 0 0 1 6 】

ここでは構造化文書として XML ベースで記述された SVG を取り上げ、スキャナ 1 0 1 によって読み取った画像をベクトルデータである SVG 形式で保存し、そのベクタ画像から指定範囲の画像を抽出する例について説明する。

20

## 【 0 0 1 7 】

図 7 (A) (B) は、本実施の形態に係るベクタ画像 (A) とその SVG 形式の構造化文書 (B) の一例を示す図である。

## 【 0 0 1 8 】

図 7 (A) に示すベクタ画像には、図形や文字、イメージのオブジェクトが複数含まれている。例えば、このベクタ画像から図 7 (A) の点線の矩形 7 0 0 で示した指定した領域のベクタ画像を記述した構造化文書データを新たに作成する例を説明する。図 7 (A) のベクタ画像は、オブジェクトとして、図形 7 0 1、文字 7 0 2、ラスタ画像 7 0 3 を有している。なお、ラスタ画像 7 0 3 はもちろんベクタ画像として直接 SVG 形式で記述することはできないが、ラスタ画像を外部参照する記述を行うことのできるため、1 つのオブジェクトとしてラスタ画像を取り上げることとする。

30

## 【 0 0 1 9 】

図 7 (B) において、構造化文書データ 7 1 0 は図形 7 0 1 の円を記述したものであり、構造化文書データ 7 1 1 は画像 7 0 3 を記述したものであり、構造化文書データ 7 1 2 は文字 7 0 2 を記述したものであり、そして構造化文書データ 7 1 3 は図形 7 0 1 の矩形を記述したものである。

## 【 0 0 2 0 】

図 8 は、例えば図 7 (A) に示すような SVG 形式で記述されたベクタ画像から、ユーザによって指定された領域 7 0 0 の画像を抽出する手順を説明するフローチャートである。

40

## 【 0 0 2 1 】

図 9 は、図 8 のステップ S 1 0 4 の各オブジェクトの処理手順を説明するフローチャートである。

## 【 0 0 2 2 】

まず図 8 において、ステップ S 1 0 2 で、CPU 2 0 1 は、図 7 (B) のベクタ画像の構造化文書データからオブジェクト化した構造化データを生成して RAM 2 0 3 に保存する。この構造化データは図 1 0 に示す DOM 形式のオブジェクトデータである。

## 【 0 0 2 3 】

図 1 0 は、図 7 (A) に示すベクタ画像に配置されるオブジェクトを DOM 形式のデー

50

タとして構造的に表した図である。

#### 【 0 0 2 4 】

DOM形式のデータをメモリに保存する際に、CPU 201は、構造化文書データに記述されているオブジェクトの色や座標のデータも同様にRAM 203に保存する。なお、CPU 201は、DOM形式のデータを生成しないで、構造化文書データが記述してあるファイルを直接読み込むことで処理する場合も考えられる。尚、図10において、「circle」は、図7(A)の円に、「image」は図7(A)の画像703に、「text」は文字702に、そして「g」に属している「rect」はそれぞれ図形701の矩形に対応している。

#### 【 0 0 2 5 】

次にステップS103で、CPU 201は、図7(A)に示す領域700のように、オペレータが操作パネル206を用いて領域を指定したことを検出する。そしてCPU 201は、指定された領域のベクタ画像内の座標位置に関する情報(具体的には左上の座標位置情報および領域の高さと幅に関する情報)を取得する。次にステップS104で、CPU 201は、DOM形式のデータとなった各図形に対して図9に示す処理を行う。

#### 【 0 0 2 6 】

図9では、まずステップS202で、CPU 201は、指定領域700にオブジェクトの全部又は一部分が含まれるか否かを判定する。この判定は、オペレータが指定した領域700の座標位置情報とベクタ画像上に配置されているオブジェクトの座標位置情報に基づいて判定される。そのCPU 201は、ここで指定された領域700内に含まれないと判定したオブジェクトを、図10のDOM形式のデータから削除する。図11は、変更後のDOM形式のデータである(S203)。

#### 【 0 0 2 7 】

図7(A)の例では、図形701の円と、一番右上の矩形が領域700に含まれていない。そのため、図11では、図形701の円に対応する「circle」と右上の矩形に対応する「rect」とが図10から除去されて図11に示されている。

#### 【 0 0 2 8 】

ステップS204では、CPU 201は、指定領域内にオブジェクトの全部が含まれていると判定するとそのまま処理を終了する。一方、全部ではなく一部分だけが含まれている場合はステップS205に処理が進む。ステップS205では、CPU 201は、そのオブジェクトが図形のオブジェクトか否かを判定する。ここで図形のオブジェクトであればステップS207に処理が進み、CPU 201は、そのオブジェクトの指定領域内に含まれる形状に、そのオブジェクトを加工する。図7(A)の例では、図形701の矩形の一部が指定領域700に含まれている。

#### 【 0 0 2 9 】

図12(A)(B)及び図13(A)(B)は、本実施の形態に係る図形オブジェクトの加工例を説明する図である。

#### 【 0 0 3 0 】

図12(A)は、図形オブジェクト1200と指定領域1201との関係を示す図、図12(B)は、この図形のオブジェクト1200を記述した構造化文書データを示している。

#### 【 0 0 3 1 】

ここで、この図形1200は、座標位置情報として、点(100, 100)、点(200, 300)、点(300, 100)、点(400, 300)、点(500, 100)、点(600, 300)を結ぶ線分で構成されたオブジェクトを示している。

#### 【 0 0 3 2 】

図13(A)(B)は、この図形オブジェクト1200を指定領域1201で抽出した例を示している。

#### 【 0 0 3 3 】

図13(B)は、指定された領域内に含まれる図形1202を記述した構造化文書デー

10

20

30

40

50

タを示している。ここでは、指定された領域を基準とした座標位置情報などを用いて、点(0, 200)、点(50, 100)、そして点(150, 300)を結ぶ線で表されている。例えば、図13(A)における点Aの座標は、図形オブジェクト1200と指定領域1201の相対関係からそれらの座標位置情報を用いることにより求めることができる。点Aは指定領域の左上の座標(450, 0)を基準としているため、点AのX座標は0となる。なお、点AのY座標は200となる。

【0034】

図14は、四角形の図形オブジェクトの加工例を示す図である。

【0035】

図14に示すように、矩形の図形オブジェクト1400の内、指定領域1401に含まれる図形1402が抽出される。ここで、図14における点Aと点Bの座標は、ベクタ画像内で図形オブジェクト1400が配置されている座標位置情報と、指定領域1401の座標位置情報とを用いることにより求められる。そして、これら点Aと点Cとをつないだ直線と、直線CDと、直線DBの3本の直線を形成することにより、抽出された図形のオブジェクト1402の形状を求められる。

【0036】

なお、図14に示す図形オブジェクト1400のSVG形式の記述は、

<svg>

```
<rect x="10" y="10" width="100" height="30" stroke="black"
stroke-width="10" fill="none" />
```

</svg>

であり、抽出された図形オブジェクト1402のSVG形式の記述は、

<svg>

```
<polyline fill="none" stroke="black" stroke-width="10"
points="0,60 30,60 30,90 0,90" />
```

</svg>

となる。

【0037】

なお、図14に示す例では、ベクタ画像内における指定領域1401の左上の座標位置は(80, -40)であり、幅130, 高さ130としている。

【0038】

図15は、円の図形オブジェクトの加工例を説明する図である。

【0039】

図15のように、円の図形オブジェクト1500のうち、指定領域1501に含まれる形状を部分的に抽出したオブジェクトが作成される。ここで図15における点Aと点Bの座標は、図形オブジェクト1500を描画している座標と、指定領域1501の座標とを用いることにより求められる。そして、これら点Aと点Bとを端点とし、図15のように円オブジェクト1500の中心と反対側の外側方向の方向点をもつ3次のベジェ曲線1502が形成される。こうして円オブジェクト1500の形状の内、指定領域1501で抽出された図形の形状を求めることができる。

【0040】

なお、図15に示す図形オブジェクト1500のSVG形式の記述は、

<svg>

```
<circle fill="none" stroke="black" stroke-width="10" cx="144.5"
cy="147" r="144" />
```

</svg>

であり、抽出された図形オブジェクト1502のSVG形式の記述は、

<svg>

```
<path fill="none" stroke="black" stroke-width="10"
d="M185, 247c0-79.5-64.5-144-144-144c-15.7, 0-30.8, 2.5-45, 7.2" />
```

10

20

30

40

50

< /svg >

となる。

【 0 0 4 1 】

なお、図 1 5 に示す例では、ベクタ画像内における指定領域 1 4 0 1 の左上の座標位置は ( 8 0 , - 1 0 0 ) であり、幅 2 0 0 , 高さ 2 0 0 としている。

【 0 0 4 2 】

ステップ S 2 0 5 で、C P U 2 0 1 は、オブジェクトが図形のオブジェクトではないと判定するとステップ S 2 0 6 に処理が進む。そして、ステップ S 2 0 6 において、C P U 2 0 1 は、文字のオブジェクトを含むか否か判定する。ここでもし C P U 2 0 1 は、文字のオブジェクトを含むと判定した場合ステップ S 2 0 9 に処理が進む。ステップ S 2 0 9 において、C P U 2 0 1 は、そのオブジェクトの指定領域内に含まれる形状にオブジェクトを加工する。

【 0 0 4 3 】

図 1 6 は、文字オブジェクト加工を説明する図である。

【 0 0 4 4 】

図 1 6 において、S V G フォントで描かれた文字オブジェクト 1 6 0 0 の内、指定領域 1 6 0 1 に含まれる文字部分のオブジェクトを抽出する。

【 0 0 4 5 】

なお、図 1 6 に示す文字オブジェクト 1 6 0 0 の S V G 形式の記述は、

< svg >

```
< path fill= " none " stroke= " black " stroke-width= " 10 "
d= " M93, 137C189 95, 155, 3.5, 78, 8C18, 12-17.5, 83, 30, 119.5
c11, 8, 33-6, 43-47.5c10-42, 6-60, 6-60 " / >
```

< /svg >

である。

【 0 0 4 6 】

図 1 7 は、この文字部分の抽出処理を説明する図である。

【 0 0 4 7 】

ここで文字は複数のベジェ曲線で構成されている。従って、前述した図形の場合と同様に、文字を描画している線と指定領域との交点を求め、それら交点との間をベジェ曲線で近似することにより、指定領域 1 6 0 1 の含まれる文字の一部分が、近似ベジェ曲線した新しいオブジェクトとして生成される。また文字のオブジェクトは、文字コードも含んでいる。ここでオブジェクトの抽出後も文字コードを保持したいならば、ユーザは、文字オブジェクトの加工を行わないという選択もできる。

このようにして抽出された文字部分のオブジェクトの S V G 形式の記述は、

< svg >

```
< path fill= " none " stroke= " black " stroke-width= " 10 "
d= " M0, 58c69, 4.5, 95, 89, 4, 129.5C94.2, 147, 69, 63, -1, 58 " / >
```

< /svg >

となる。

【 0 0 4 8 】

なお、指定領域の左上の座標位置は ( 8 9 , - 5 0 ) であり、幅 3 0 0 , 高さ 3 0 0 である。

【 0 0 4 9 】

また文字のオブジェクトが文字列の場合であれば、C P U 2 0 1 は、その文字列に含まれる 1 つ 1 つの文字を描画している座標を用いて指定領域に含まれるか否か判定して、指定領域内に入る文字列を抽出する。例えば図 7 ( A ) の文字列「 A B C D E F 」において、文字列「 D E F 」が指定領域 7 0 0 の外になる場合は、構造化文書データの「 < text > ABCDEF < / text > 」は、「 < text > ABC < / text > 」に変換される。

【 0 0 5 0 】



ステップS206で、CPU201は、オブジェクトが文字のオブジェクトではないと判定するとステップS208に処理が進む。ステップS208において、CPU201は、ラスタ形式の画像オブジェクトか否か判定する。ここでCPU201が、ラスタ形式の画像オブジェクトでないと判定した場合、処理が終了する。ラスタ形式の画像オブジェクトが存在するとCPU201によって判定された場合、ステップS210に処理が進む。ステップS210において、CPU201は、そのオブジェクトの指定領域内に含まれる形状にオブジェクトを加工する。このイメージデータの加工処理において、例えば後述する図19に示すように、CPU201は、図7のイメージ703のラスタ形式の画像オブジェクトに対応するイメージファイル「image.png」のデータの内、指定領域700に含まれる右半分の幅「50」の画像オブジェクトに対応する画像データを抽出する。そして、CPU201は、新たなイメージファイルを作成して別ディレクトリに保存する処理を行う。このとき、元のイメージファイルと同じディレクトリに新たなイメージファイルを保存する場合、新たなイメージファイルのファイル名称が「image1.png」などと変更される必要がある。

10

#### 【0051】

このようにして、指定した領域内のイメージのみを切り出すことにより、本実施の形態における処理は、その指定領域内のオブジェクトを記憶する記憶容量を削減できる。また画像データのビットマップ等データ量が大きい場合には、解像度変換を行うことによりデータ量を小さくすることもできる。

#### 【0052】

20

尚、上述した図形、文字、およびイメージのオブジェクト加工処理において、オブジェクトを加工する前の各オブジェクトのデータ量がRAM203に保存しておいてもよい。そして、この各オブジェクトのデータ量を参照し、オブジェクトの種類によって加工前よりも加工後の方が構造化文書のデータ量が増加する場合には、ユーザがそのオブジェクトの加工を行わないように選択できる設定画面を設けることができる。

#### 【0053】

また、オブジェクトの加工処理に時間がかかる場合には、ユーザは、そのオブジェクトの加工を行わないように選択できるようにしても良い。

#### 【0054】

図18は、他のオブジェクトの加工例を説明する図である。

30

#### 【0055】

加工前よりも加工後のオブジェクトのデータ量が増加してしまう例として、3次ベジェ曲線のオブジェクトが考えられる。CPU201は、前述のような指定された領域内のみを残したオブジェクトの加工処理を行なうことにより、ベジェ曲線のオブジェクトの一部として、点C、Dの区間のベジェ曲線のオブジェクトを生成する。この場合、データ量が増えてしまい、加工処理においても時間がかかる場合があると考えられる。そこで、CPU201は、点B、D、Cの区間のベジェ曲線のオブジェクトを生成する。この処理によって、少ないデータ量でオブジェクトの一部分の抽出が可能となる。

#### 【0056】

こうして図8のステップS104における処理が完了すると、ステップS105に進み、CPU201は、DOMが所定の仕様を満たしているか否か判定する。ここでもしDOMが所定の仕様を満たしていないと判定するとステップS107に進み、仕様を満たすように不足している記述や不要な記述を編集する。またステップS105で、DOMが所定の仕様を満たしている場合、或はステップS107を実行するとステップS106に進み、CPU201は、その編集した構造化文書データを書き出す。

40

#### 【0057】

図19は、元のベクタ画像から一部のベクタ画像を抽出した構造化文書データの一例を示す図である。

#### 【0058】

図19における1900は、図7(A)に示すオブジェクトに対応する構造化文書デー

50

タである。

【 0 0 5 9 】

ここでは図 7 ( A ) に示すように、指定領域 7 0 0 でイメージ 7 0 3 の一部 ( 右半分 : 座標 ( 1 5 0 , 5 0 0 ) から幅 5 0 、高さ 1 0 0 のイメージ ) と文字列の「 A B C 」が抽出されたオブジェクトデータとして、図 1 9 の構造化文書データ 1 9 0 1 が示される。このオブジェクト抽出後の構造化文書データ 1 9 0 1 において、オブジェクトをグループ化する機能をもつグループ要素 < g fill = “ blue ” > は 1 つも子要素を持っていない。すなわち、グループ要素の子要素が存在しなくなった場合、グループ要素の記述が不要となるため、削除する処理を実行してもよい。

【 0 0 6 0 】

そしてステップ S 1 0 8 で、CPU 2 0 1 は、その構造化文書データを外部機器 ( プリント等 ) に転送して処理を終了する。ここで LAN 1 0 2 に接続されているネットワークプリンタ 1 0 4 、ファイルサーバ 1 0 5 等の外部機器に対して、その抽出したベクタ画像を転送する。プリンタやスキャナの処理能力により 1 ページ全体の画像の転送が不可能なときには、転送可能なデータ容量となるように範囲を決め順次抽出して転送することにより、画像の全体の転送が可能となる。

【 0 0 6 1 】

尚、本実施の形態では、PC 1 0 3 を用いて処理を実行する例を示したが、スキャナ 1 0 1 における処理においても本実施の形態を適用することができる。この場合、スキャナ 1 0 1 に読み込まれた画像をベクタ画像化する。その後上述した処理を同様に行うことになる。

【 0 0 6 2 】

また、オペレータが操作パネルで指定する領域は、矩形に限らず円やフリーハンドで描いた領域であっても良い。またタッチパネルにより領域を指定しても良い。

【 0 0 6 3 】

また、プリンタやコピー機、FAX等の機器に転送する際には、それら機器の処理能力に応じた指定領域の選択を、オペレータ又は機器の制御部が行う場合が考えられる。

【 0 0 6 4 】

以上説明した処理によって以下に示すような効果が得られる。例えば、SVGなどのベクタ画像で描画された地図において、ユーザが必要とする地域のみを範囲指定して切り出すことによって、ユーザが必要としない地域の地図情報を削除する。その後、ユーザは切り出した地図に対応する構造化文書データをリソースの少ない携帯電話などに転送しても描画することが可能となり、その描画処理もより短い時間で可能になる。

【 0 0 6 5 】

[ 実施の形態 2 ]

図 2 3 は、小さくて見づらい文字や図形、画像を拡大表示し、その拡大状態を保持してデータの複製を行う手順を説明するフローチャートである。

【 0 0 6 6 】

例えば、図 2 0 ( A ) に示すウィンドウ内の描画領域のうち所望の矩形領域を指定すると、図 2 0 ( B ) に示すように、その指定された矩形領域内のオブジェクトを拡大表示する処理が知られている。この場合、前述の実施の形態 1 に係る処理を実行すると、図 2 1 に示すような描画処理が実行される構造化文書データが得られる。本実施の形態 2 は、この実施の形態 1 の発展形として、図 2 0 ( B ) に示すように、指定された矩形領域内のオブジェクトの拡大を維持した構造化文書に変換するものである。このような処理を用いることによって、あるユーザが友人に画像の一部を電子メールなどで知らせたい場合、拡大画像を適切に提示することができる。

【 0 0 6 7 】

まず、図 2 2 ( A ) のように、ユーザがマウスを用いることによって、拡大表示したい領域を指定する。CPU 2 0 1 は、この指定された領域を検出する ( S 3 0 1 )。次に、CPU 2 0 1 は、この指定を検出し、図 2 2 ( B ) に示すようにウィンドウ上で指定され

10

20

30

40

50

た領域を拡大表示する（S 3 0 2）。図 2 2（B）は、図 2 2（A）を部分的に拡大表示した地図である。拡大表示される地図情報の領域には、文字データや図形データ、画像データなどが含まれる。

【 0 0 6 8 】

次に CPU 2 0 1 は、ユーザがマウスを用いてその拡大された矩形領域内の描画データの保存の指示を実行したことを検出する（S 3 0 3）。この操作の検出に応じて、CPU 2 0 1 は、ウィンドウ上で拡大表示している矩形領域の情報として、図 2 2（A）の領域を基準とした左上の座標（3 4 0, 5 9 4）と、幅 2 3 0 p x、高さ 2 3 0 p x を RAM 2 0 3 に保存する（S 3 0 4）。また、ユーザが指定した拡大表示倍率に関する情報（2 倍）を RAM 2 0 3 に保存する（S 3 0 5）。尚、実際には、図 2 2（B）に示す地図の図 2 2（A）に示す地図の表示倍率の 2 倍ではないが、本実施の形態 2 では、2 倍であるものとして説明する。

10

【 0 0 6 9 】

次に、ユーザがマウスなどのポインティングデバイスを用いることによって、選択したデータを RAM 2 0 3 に保存する方法について説明する。

【 0 0 7 0 】

図 2 5 は、図 2 2（A）に示すオブジェクトを記述した構造化文書データを示す図である。図 2 5 の構造化文書データは、標準化団体 W 3 C で仕様が策定されている Scalable Vector Graphics（SVG）を用いて記述したものである。

【 0 0 7 1 】

20

図 2 5 に示す構造化文書データは、各都道府県に関する地図情報を有する。具体的には、都道府県名や市町村名を表す文字オブジェクトや、都道府県や市町村の形状を表すパスオブジェクトデータ（輪郭データ）を含む。なお、画像データに関するイメージオブジェクトを用いることによって、都道府県や市町村の形状を表す場合も考えられる。更に、幹線道路や鉄道の路線などの情報を示すパスオブジェクトデータを有してもよい。

【 0 0 7 2 】

そして、図 2 2（A）に示す地図を描画するにあたり、図 2 5 の構造化文書データに対応する W 3 C で仕様が定められた DOM ツリーが RAM 2 0 3 に記憶されているが、CPU 2 0 1 は、この DOM のツリーを参照する（S 3 0 6）。

【 0 0 7 3 】

30

図 2 6 は、図 2 5 の構造化文書データから生成された DOM ツリーを示す図である。

【 0 0 7 4 】

そして、CPU 2 0 1 は、図 2 6 の日本全体の地図に関する DOM ツリーと同じものを RAM 2 0 3 の内部に複製する（S 3 0 7）。次に CPU 2 0 1 は、その複製して得られた DOM ツリーから、ユーザが指定した領域である東京都に関する地図データのノードを抽出するとともに、その他の全ての都道府県に関する地図データのノードを削除する（S 3 0 8）。更に CPU 2 0 1 は、その複製して得られた DOM ツリーにおいて、拡大表示の矩形領域に一部分が含まれる文字や図形、画像の各オブジェクトのノードを抽出して、抽出されたノードに対応するオブジェクトの加工を行う（S 3 0 9）。

【 0 0 7 5 】

40

図 2 4 は、図 2 3 のステップ S 3 0 9 の処理の詳細な処理手順を示すフローチャートである。図 2 5 の構造化文書データ内の < g > 要素の各子要素のオブジェクトについて、それぞれ以下の処理を実行する。

【 0 0 7 6 】

まず CPU 2 0 1 は、あるオブジェクトに対して、ユーザが設定した矩形領域内に全部含まれるかどうか判断する（S 4 0 1）。もし、矩形領域に全部のオブジェクトが含まれていると判断したならば、CPU 2 0 1 は、すべてのオブジェクトを削除せずに処理を終了する。もし、矩形領域に全部含まれないと判断したならば、CPU 2 0 1 は、対象のオブジェクトが文字のオブジェクトであるかどうか判断する（S 4 0 2）。ここでもし、文字のオブジェクトであると判断したならば、CPU 2 0 1 は、その文字のオブジェクトの

50

うち矩形領域に入らない文字列を削除して、文字オブジェクトの加工を行う（S405）。一方、ステップS402で、文字のオブジェクトでないと判断した場合はステップS403に進み、CPU201は、図形のオブジェクトであるかどうか判断する。ここで図形のオブジェクトであると判断するとステップS406で、CPU201は、その図形オブジェクトのうち矩形領域に含まれる部分を抽出してRAM203に保存し、図形オブジェクトを更新することによって加工を行う。

【0077】

またステップS403で、図形のオブジェクトでないと判断した場合はステップS404に進み、CPU201は、画像のオブジェクトであるかどうか判断する。ここで画像のオブジェクトであると判断するとステップS407で、CPU201は、加工処理として、その画像オブジェクトのうち矩形領域に含まれる画像データを抽出してRAM203に保存する。そしてCPU201は、画像オブジェクトを更新する。

【0078】

これらステップS401～S407の処理をCPU201は、各オブジェクトについて実行する。

【0079】

図27は、ユーザが指定した矩形領域に含まれるオブジェクトの加工を行うことによって得られた構造化文書データで表した図である。また、図28は、図27の東京都のみの地図データを含む構造化文書データをDOMツリーで表した図である。

【0080】

この一連の構造化文書データ（DOMツリー）に含まれるオブジェクトの加工処理によって、ユーザが選択した東京都以外に関するオブジェクトは削除される。そして、矩形領域からはみ出していた文字や図形、画像オブジェクトの部分が除かれるように更新される。尚、携帯機器などリソースが限られる機器においては、図23のステップS309の処理の変形例として、オブジェクトの加工を行わずに、矩形領域に一部分でも含まれている場合には、複製したDOMから削除しない処理を実行してもよい。

【0081】

尚、図24のステップS401における矩形領域と各文字オブジェクトとの内外判定は、以下の通りである。即ち、図25に示される文字オブジェクトの座標位置およびフォントのサイズ（および種類）より画素単位で決定される各描画位置が矩形領域に含まれるかどうかを判定することにより決定される。同様に、図形のオブジェクト、画像のオブジェクトに関しても、構造化文書データ内のオブジェクトデータに基づいて決定される画素単位で決定される描画位置が矩形領域に含まれるかどうかを判定することにより決定される。

【0082】

再び、図23に示す処理の説明に戻る。前述のステップS309に続いてCPU201は、図27に示す東京都の地図データが図22（B）に示す拡大表示時の地図データとして描画される構造化文書データを取得するための処理を行う。CPU201は、図27の構造化文書データを、図22（B）の拡大表示倍率に関する情報を含む構造化文書データに変換する（S310）。この1つ目の方法として、アフィン変換を行うことによって前記構造化文書データを更新する方法が考えられる。

【0083】

まず、RAM203に保存された拡大表示の矩形領域の左上の座標（340，594）と幅230px、高さ230pxに基づいて、その矩形領域の中心位置を原点座標（0，0）に重なるように平行移動する。そして、内部メモリに保存している拡大倍率2倍に基づいて拡大し、拡大した矩形領域の左上の座標が原点（0，0）になるように平行移動する。

【0084】

以上の一連の手続きを示す属性値は、

「transform=「translate(230, 230), scale(2, 2), translate(-455, -709)」」

10

20

30

40

50

である。このtransform属性をsvg要素の子要素であるg要素の属性として追加する。

#### 【0085】

この属性値を一般化すると、拡大表示の際に地図の領域内においてユーザが指定した矩形領域の左上の座標を $(x, y)$ 、幅を $w$ 、高さを $h$ とし、拡大倍率を $v$ とすると、transform属性の属性値は、

「 $\text{translate}(w/2 * v, h/2 * v), \text{scale}(v, v), \text{translate}(-x-w/2, -y-h/2)$ 」

となる。そして、このtransform属性をg要素に付加することによって、図27に示す構造化文書データが図29に示す構造化文書データに変換される。

#### 【0086】

また、第2の方法として、地図データが描画される初期の中心位置を、拡大表示領域に合わせるアフィン変換を用いた方法が考えられる。具体的には、SVGにおけるsvg要素に、空白又はコンマで区切られた4個の数値のリストの属性値を有するviewBox属性データを追加する処理が実行される。

#### 【0087】

まず、CPU201は、地図領域に対する、RAM203に保存された拡大表示される矩形領域の左上の座標 $(340, 594)$ をviewBox属性の属性値における最初の2つの数値として指定する。そして、拡大表示領域の幅 $230\text{px}$ と高さ $230\text{px}$ に対して、RAM203に保存している拡大倍率2を掛けることによって求めたものを、残り2つの数値リストとして指定する。即ち、「 $\text{viewBox} = "340\ 594\ 460\ 460"$ 」のviewBox属性をsvg要素に追加することになる。

#### 【0088】

一般的には、拡大表示の際にユーザの設定した矩形領域の左上の座標を $(x, y)$ 、幅を $w$ 、高さを $h$ とし、拡大倍率を $v$ とすると、viewBox属性の属性値は

「 $x\ y\ w * v\ h * v$ 」

となる。そして、このviewBox属性をsvg要素に付け加えることによって、図30のような構造化文書データとなる。

#### 【0089】

第3の方法として、文字オブジェクトのフォントサイズと位置座標、また、パスオブジェクト(輪郭オブジェクト)の $x, y$ 座標に関するd属性の属性値を更新する方法が考えられる。

#### 【0090】

まず、文字オブジェクトの更新方法について説明する。フォントサイズは、RAM203に保存された拡大表示倍率2倍に基づいて、フォントサイズが2倍とされる。また、文字オブジェクトの位置座標は、地図領域を基準とした拡大表示される矩形領域の左上の座標 $(340, 594)$ から地図領域を基準とした文字オブジェクトの位置情報の差分によって求められる。すなわち、文字オブジェクトについては、

$\text{< text } x = "113" \ y = "126" \ \text{font-size} = "20" > \text{Tokyo} < / \text{text} >$

となる。

#### 【0091】

次に、東京都の輪郭を表すパスオブジェクトの更新方法について説明する。まず、東京都の形状を表すパスオブジェクトを含む拡大表示枠の矩形領域の中心位置を原点座標 $(0, 0)$ に重なるように並行移動する。そして、RAM203に保存された拡大倍率2倍に基づいて拡大し、拡大した矩形領域の左上の座標が原点 $(0, 0)$ になるように平行移動する。即ち、東京都の輪郭を表すd属性の属性値については、前記アフィン変換を用いた手段におけるtransform属性を用いた手段と同様の変換を用いた計算が行われる。また、パスオブジェクトの線の太さに関する属性stroke-widthの属性値については、拡大倍率2倍に基づいて求めることができる。よって、パスオブジェクトは、

$\text{< path } \text{fill} = "none" \ \text{stroke} = "black" \ \text{stroke-width} = "4.0"$

$d = "M\ 118.1902\ 206.2466\ C\ 71.6528\ 206.2466\ \dots (略) \dots" / >$

となる。このようにして図27の構造化文書データから変換された構造化文書データを図

10

20

30

40

50

31に示す。

【0092】

以上の操作によって、予め拡大表示された小さくて見づらい地図や図面などの描画データを保存操作する際に、拡大表示状態を保持した構造化文書データを変換しているので、拡大表示状態を維持した描画データの保存を行うことが可能となる。

【0093】

なお本発明は、前述した実施の形態の機能を実現するソフトウェアのプログラムを、システム或いは装置に直接或いは遠隔から供給し、そのシステム或いは装置のコンピュータが、その供給されたプログラムコードを読み出して実行することによっても達成される場合を含む。その場合、プログラムの機能を有していれば、その形態はプログラムである必要はない。従って、本発明の機能処理をコンピュータで実現するために、該コンピュータにインストールされるプログラムコード自体も本発明を実現するものである。つまり、本発明には、本発明の機能処理を実現するためのコンピュータプログラム自体も含まれる。その場合、プログラムの機能を有していれば、オブジェクトコード、インタプリタにより実行されるプログラム、OSに供給するスクリプトデータ等、プログラムの形態を問わない。

【0094】

プログラム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、MO、CD-ROM、CD-R、CD-RW、磁気テープ、不揮発性のメモ리카ード、ROM、DVD（DVD-ROM、DVD-R）などがある。その他のプログラムの供給方法としては、クライアントコンピュータのブラウザを用いてインターネットのホームページに接続し、該ホームページから本発明のコンピュータプログラムそのもの、もしくは圧縮され自動インストール機能を含むファイルをハードディスク等の記憶媒体にダウンロードすることによっても供給できる。また本発明のプログラムを構成するプログラムコードを複数のファイルに分割し、それぞれのファイルを異なるホームページからダウンロードすることによっても実現可能である。つまり本発明の機能処理をコンピュータで実現するためのプログラムファイルを複数のユーザに対してダウンロードさせるWWWサーバも、本発明のクレームに含まれるものである。

【0095】

また、本発明のプログラムを暗号化してCD-ROM等の記憶媒体に格納してユーザに配布し、所定の条件を満足するユーザに対してインターネットを介してホームページから暗号化を解く鍵情報をダウンロードさせ、その鍵情報を使用することにより暗号化されたプログラムを実行してコンピュータにインストールさせて実現することも可能である。

【0096】

またコンピュータが、読み出したプログラムを実行することによって、前述した実施形態の機能が実現される他、そのプログラムの指示に基づき、コンピュータ上で稼動しているOSなどが、実際の処理の一部又は全部を行ない、その処理によっても前述した実施形態の機能が実現され得る。

【0097】

さらに、記録媒体から読み出されたプログラムが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部又は全部を行ない、その処理によっても前述した実施形態の機能が実現される。

【図面の簡単な説明】

【0098】

【図1】図形、文字、イメージデータのオブジェクトを含むベクタ画像画像の一例を示す図である。

【図2】領域を指定して図形オブジェクトの一部を抽出する例を示す図である。

- 【図 3】領域を指定して文字オブジェクトの一部を抽出する例を示す図である。
- 【図 4】領域を指定してイメージオブジェクトの一部を抽出する例を示す図である。
- 【図 5】本発明の実施の形態に係るシステムの概要を示した構成図である。
- 【図 6】本実施の形態に係る P C 1 0 3 の構成を示すブロック図である。
- 【図 7】本実施の形態に係るベクタ画像 ( A ) とその構造化文書データ ( B ) の一例を示す図である。
- 【図 8】本実施の形態において、ベクタ画像から、指定した領域の画像を抽出する手順を説明するフローチャートである。
- 【図 9】図 8 のステップ S 1 0 4 の各オブジェクトの処理手順を説明するフローチャートである。
- 【図 1 0】図 7 ( A ) に示す画像を D O M 形式で表した図である。
- 【図 1 1】図 1 0 の D O M から指定領域に含まれないオブジェクトを削除した例を示す図である。
- 【図 1 2】、
- 【図 1 3】本実施の形態に係る図形オブジェクトの加工例を説明する図である。
- 【図 1 4】四角形の図形オブジェクトの加工例を説明する図である。
- 【図 1 5】円の図形オブジェクトの加工例を説明する図である。
- 【図 1 6】文字オブジェクト加工を説明する図である。
- 【図 1 7】文字オブジェクトの抽出処理を説明する図である。
- 【図 1 8】他のオブジェクトの加工例を説明する図である。
- 【図 1 9】元のベクタ画像から一部のベクタ画像を抽出した構造化文書データの一例を示す図である。
- 【図 2 0】描画データの拡大表示処理を説明するための図である。
- 【図 2 1】描画データの表示処理を説明するための図である。
- 【図 2 2】描画データの拡大表示処理を説明するための図である。
- 【図 2 3】本発明の実施の形態 2 に係る矩形領域内の描画データを拡大表示させるための構造化文書データの変換処理を示すフローチャートである。
- 【図 2 4】図 2 3 のステップ S 3 0 9 の処理の詳細な処理手順を示すフローチャートである。
- 【図 2 5】図 2 2 ( A ) に示すオブジェクトを記述した構造化文書データを示す図である。
- 【図 2 6】図 2 5 の構造化文書データから生成された D O M ツリーを説明する図である。
- 【図 2 7】実施の形態 2 において、ユーザが指定した矩形領域に含まれるオブジェクトの加工を行うことによって得られた構造化文書データで表した図である。
- 【図 2 8】図 2 7 の構造化文書データを D O M ツリーで表した図である。
- 【図 2 9】実施の形態 2 に係る変換処理によって得られた構造化文書データの一例を示す図である。
- 【図 3 0】実施の形態 2 に係る変換処理によって得られた構造化文書データの一例を示す図である。
- 【図 3 1】図 2 7 の構造化文書データから変換された構造化文書データの一例を示す図である。

【符号の説明】

【 0 0 9 9 】

- 2 0 1 C P U ( 処理ユニット )
- 2 0 2 R O M
- 2 0 3 R A M
- 2 0 4 記憶部
- 2 0 6 操作パネル
- 2 0 7 表示部

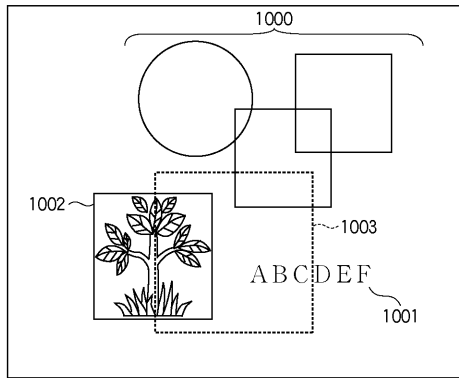
10

20

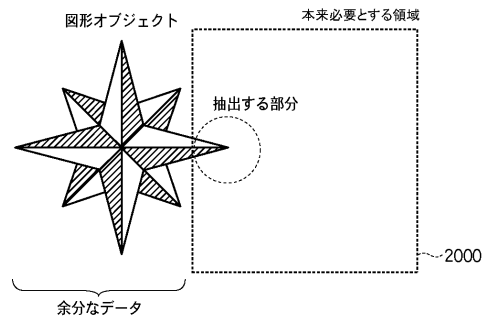
30

40

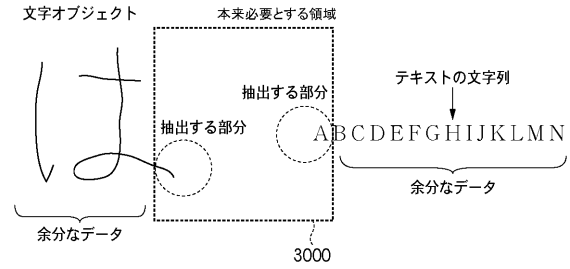
【 図 1 】



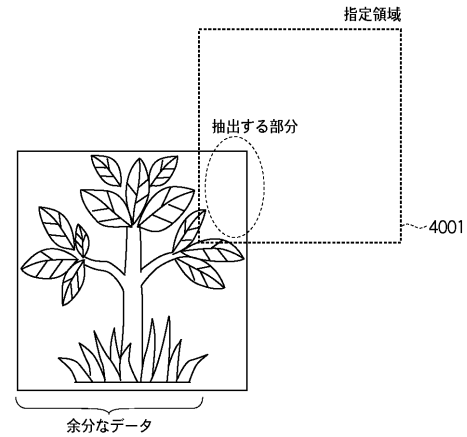
【 図 2 】



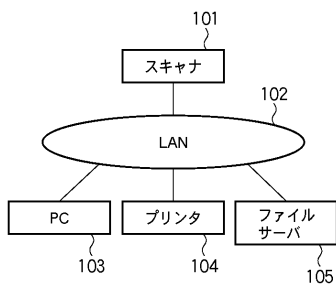
【 図 3 】



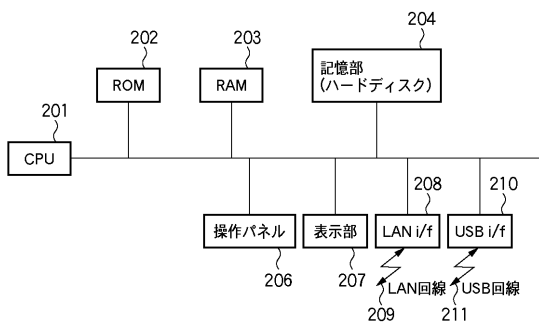
【圖 4】



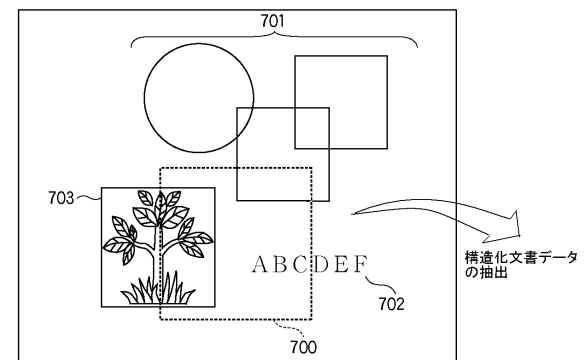
【 図 5 】



【 図 6 】



【 図 7 】

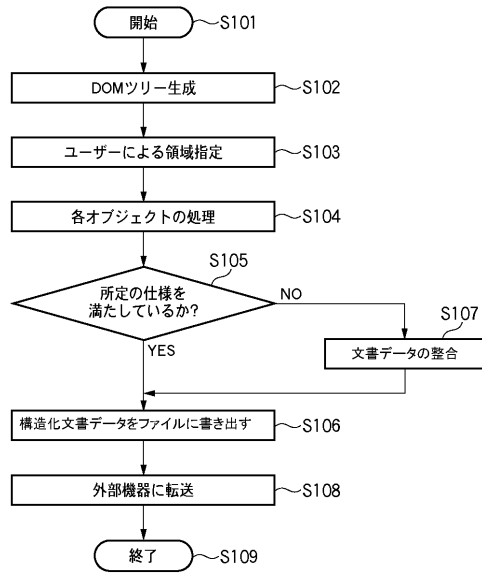


```
<svg>
  <circle cx='300' cy='100' i='100'
    stroke='blue' stroke-width='10'/> } 710
  <image x='100' y='500' width='100' height='100'
    xlink :href='image.png'/> } 711
  <text>ABCDEF</text> } 712
</>
  <rect x='400' y='200' width='100' height='100'/>
  <rect x='450' y='100' width='100' height='100'/> } 713
</svg>
```

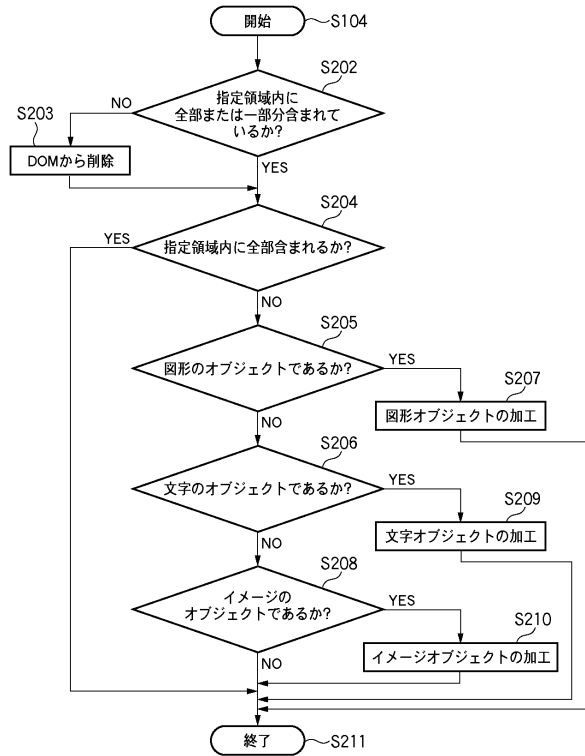
(B)



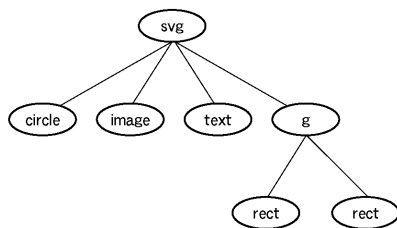
【図 8】



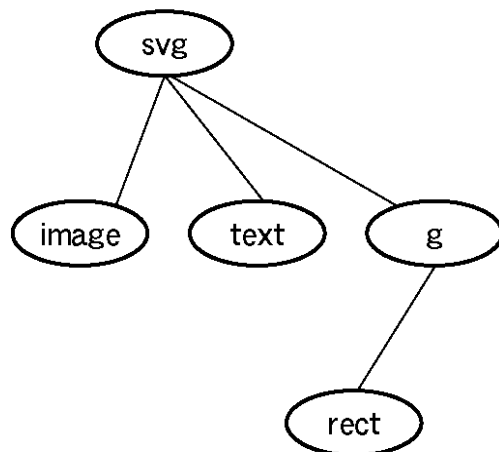
【図 9】



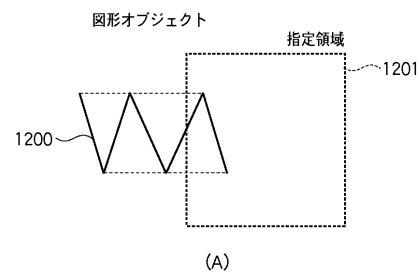
【図 10】



【図 11】



【図 12】

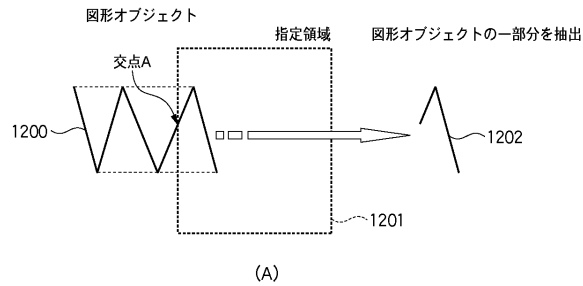


```

<svg>
  <polyline fill="none" stroke="black" stroke-width="15"
    points="100,100
            200,300 300,100 400,300 500,100
            600,300"/>
</svg>
  
```

(B)

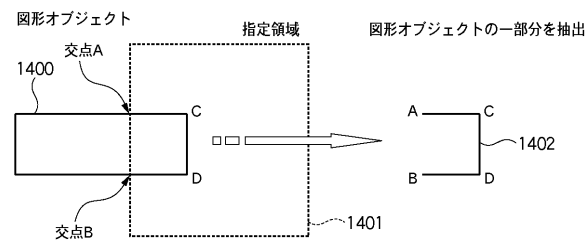
【図 13】



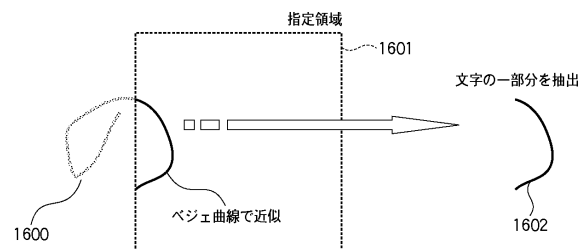
```
<svg>
<polyline fill='none' stroke='black' stroke-width='15'
  points='450,200 500,100 600,300' />
</svg>
```

(B)

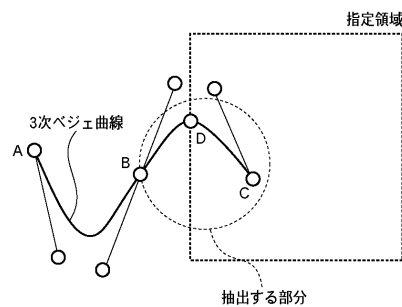
【図 14】



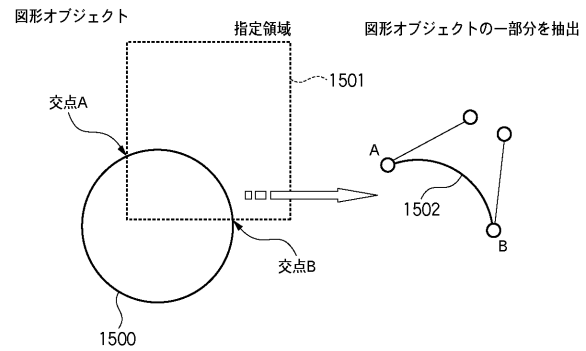
【図 17】



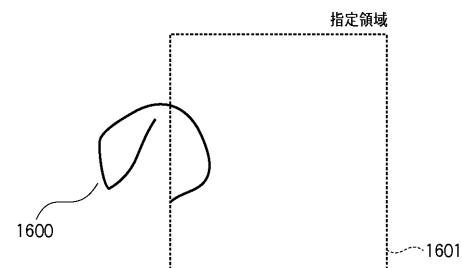
【図 18】



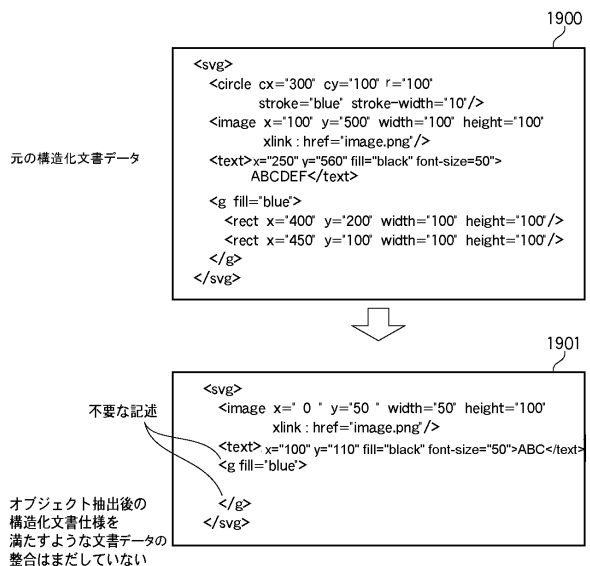
【図 15】



【図 16】



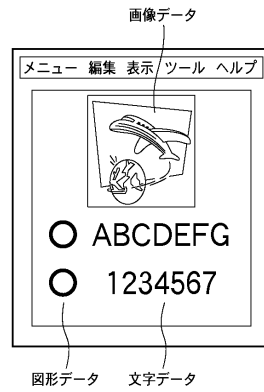
【図 19】



【図 20】

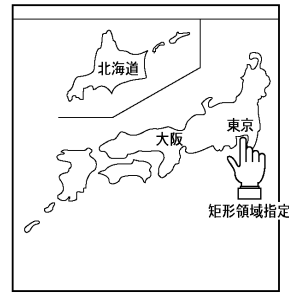


(A)



(B)

【図 22】

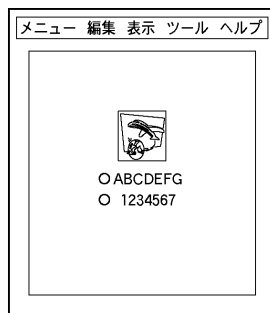


(A)



(B)

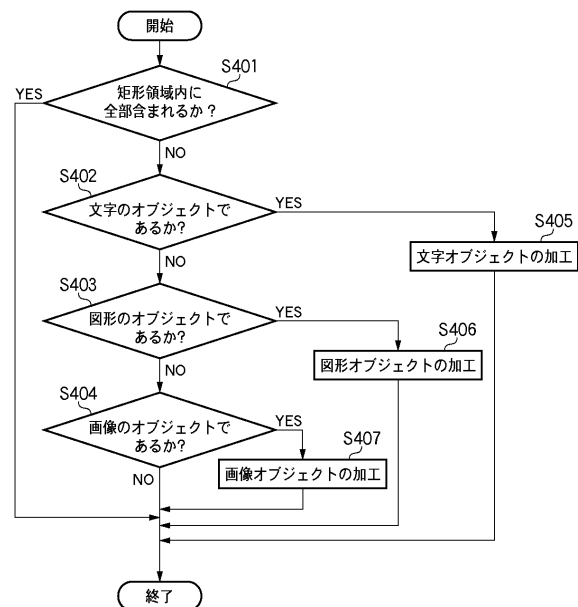
【図 21】



【図 23】



【図 24】



## 【図 25】

変換処理前の構造化文書データ

```

<?xml version='1.0' encoding='UTF-8' standalone='no'?>
<svg xmlns='http://www.w3.org/2000/svg'
  xmlns:xlink='http://www.w3.org/1999/xlink'
  x='0' y='0'
  width='1500' height='1500'>

  <g>
    <text x='76.9384995' y='58.6819153' font-size='10'>Japan</text>

    <text x='270' y='300' font-size='10'>Hokkaido</text>
    <path fill='none' stroke='black' stroke-width='2.0'
      d='M 238.6485 274.5447 C 250.5595 250.7218 ...(略)... '/>

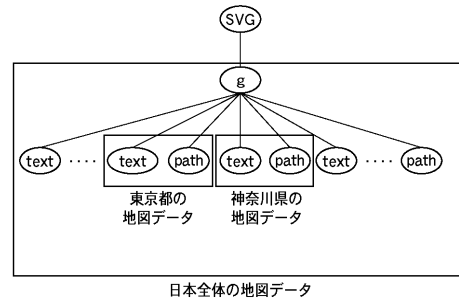
    <text x='453' y='720' font-size='10'>Tokyo</text>
    <path fill='none' stroke='black' stroke-width='2.0'
      d='M 399.0951 697.1233 C 375.8264 697.1233 ...(略)... '/>

    <text x='442' y='796' font-size='10'>Kanagawa</text>
    <path fill='none' stroke='black' stroke-width='2.0'
      d='M 350.5243 723.1831 C 357.194 725.8937 ...(略)... '/>

    :
    (略)
    :
  </g>
</svg>

```

## 【図 26】



## 【図 27】

変換処理によって得られた構造化文書データ

```

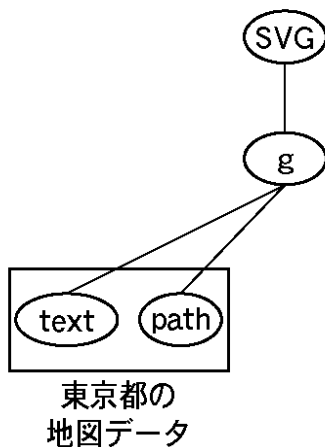
<?xml version='1.0' encoding='UTF-8' standalone='no'?>
<svg xmlns='http://www.w3.org/2000/svg'
  xmlns:xlink='http://www.w3.org/1999/xlink'
  x='0' y='0'
  width='1500' height='1500'>

  <g>
    <text x='453' y='720' font-size='10'>Tokyo</text>
    <path fill='none' stroke='black' stroke-width='2.0'
      d='M 399.0951 697.1233 C 375.8264 697.1233 ...(略)... '/>

  </g>
</svg>

```

## 【図 28】



## 【図 30】

変換処理によって得られた構造化文書データ

```

<?xml version='1.0' encoding='UTF-8' standalone='no'?>
<svg xmlns='http://www.w3.org/2000/svg'
  xmlns:xlink='http://www.w3.org/1999/xlink'
  x='0' y='0'
  width='1500' height='1500'
  viewBox='340 594 460 460'>
  <g>
    <text x='453' y='720' font-size='10'>Tokyo</text>
    <path fill='none' stroke='black' stroke-width='2.0'
      d='M 399.0951 697.1233 C 375.8264 697.1233 ...(略)... '/>
  </g>
</svg>

```

追加データ

## 【図 31】

変換処理によって得られた構造化文書データ

```

<?xml version='1.0' encoding='UTF-8' standalone='no'?>
<svg xmlns='http://www.w3.org/2000/svg'
  xmlns:xlink='http://www.w3.org/1999/xlink'
  x='0' y='0'
  width='1500' height='1500'>
  <g>
    <text x='113' y='126' font-size='20'>Tokyo</text>
    <path fill='none' stroke='black' stroke-width='4.0'
      d='M 118.1902 206.2466 C 71.6528 206.2466 ...(略)... '/>
  </g>
</svg>

```

更新したデータ

## 【図 29】

変換処理によって得られた構造化文書データ

```

<?xml version='1.0' encoding='UTF-8' standalone='no'?>
<svg xmlns='http://www.w3.org/2000/svg'
  xmlns:xlink='http://www.w3.org/1999/xlink'
  x='0' y='0'
  width='1500' height='1500'>
  <g transform='translate(230,230),scale(5.5),translate(-455,-709)'>
    <text x='453' y='720' font-size='10'>Tokyo</text>
    <path fill='none' stroke='black' stroke-width='2.0'
      d='M 399.0951 697.1233 C 375.8264 697.1233 ...(略)... '/>
  </g>
</svg>

```

追加データ

---

フロントページの続き

(51)Int.Cl. F I  
G 0 6 T 11/80 B  
G 0 6 T 11/00 2 0 0

(72)発明者 内田 均  
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 居島 一仁

(56)参考文献 特開2005-056039(JP,A)  
特開2000-242484(JP,A)  
特開2003-303091(JP,A)  
特開2001-209369(JP,A)  
特開2001-312736(JP,A)

(58)調査した分野(Int.Cl., DB名)  
G 0 9 G 3 / 0 0 - 5 / 4 2  
G 0 6 F 9 / 0 6、1 7 / 2 1