(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2006-106971
(P2006-106971A)
(43)公開日　平成18年4月20日(2006.4.20)

(51)Int.Cl.	F I	テーマコード（参考）
G06T 7/60 (2006.01)	G06T 7/60 200K	5L096

審査請求 未請求　請求項の数 10　ＯＬ　（全 28 頁）

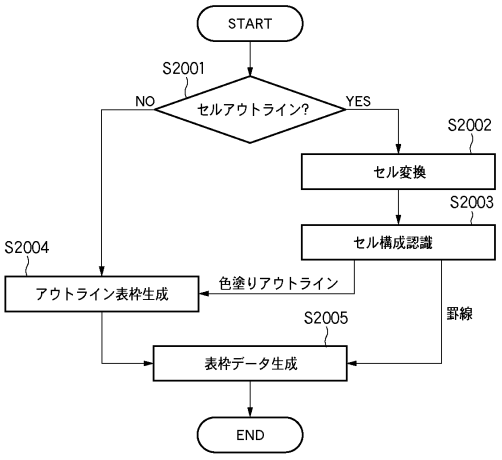
(21)出願番号　特願2004-290384 (P2004-290384)
(22)出願日　平成16年10月1日 (2004.10.1)

(71)出願人　000001007
キヤノン株式会社
東京都大田区下丸子３丁目３０番２号
(74)代理人　100076428
弁理士　大塚　康徳
(74)代理人　100112508
弁理士　高柳　司郎
(74)代理人　100115071
弁理士　大塚　康弘
(74)代理人　100116894
弁理士　木村　秀二
(72)発明者　鵜沢　充
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

Ｆターム(参考)　5L096 BA07　BA17　FA03　FA06　FA16

(54)【発明の名称】表ベクトルデータ生成方法及び文書処理装置

(57)【要約】
【課題】　スキャン画像のようなノイズの多い原稿に対して、原本を損なうことなく、原本に忠実な表枠を表現するベクトルデータを作成する。
【解決手段】　表を構成する２値データをアウトライン化したデータが表枠のセルを構成しているか否かを判定し（Ｓ２００１）、表枠のセルを構成していると判定されたデータを罫線と色塗りアウトラインで表枠を表現するデータに変換する（Ｓ２００２）。そして、その変換されたデータと表枠のセルを構成しないと判定されたデータとを合成し、表枠を表現するベクトルデータを生成する（Ｓ２００５）。
【選択図】　図２０



【特許請求の範囲】
【請求項１】
表を構成する２値データから表ベクトルデータを生成する表ベクトルデータ生成方法であって、
前記２値データをアウトライン化したデータが表枠のセルを構成しているか否かを判定する工程と、
前記判定する工程で前記表枠のセルを構成していると判定されたデータを罫線と色塗りアウトラインで表枠を表現するデータに変換する工程と、
前記変換されたデータと前記判定する工程で前記表枠のセルを構成しないと判定されたデータとを合成し、表枠を表現するベクトルデータを生成する工程と、
を有することを特徴とする表ベクトルデータ生成方法。
【請求項２】
前記変換する工程では、前記表枠のセルを構成しているデータを、外枠を構成する外枠構成データとセルを構成するセル構成データとに分割する工程と、
前記外枠構成データより前記セル構成データをマッピングするためのマッピング領域を作成する工程と、
前記セル構成データをセル図形に変換する工程と、
前記マッピング領域へ前記セル図形をマッピングし、表中の罫線構成を認識する工程と、
前記セル図形を構成するセルデータを用いて正確なセル位置及び罫線位置、線幅を抽出する工程とを有し、
前記セル図形をマッピングする際に、何もマッピングされない領域について色塗りアウトラインを生成し、前記抽出する工程により色塗りアウトラインの正確な位置を抽出することを特徴とする請求項１記載の表ベクトルデータ生成方法。
【請求項３】
前記セル図形は属性情報をもつ矩形図形であり、前記セル図形を生成する際に、該セル図形を統合、分割する工程を有し、
前記認識する工程で前記セル図形の属性情報に応じて罫線を追加、削除することを特徴とする請求項２記載の表ベクトルデータ生成方法。
【請求項４】
前記属性情報は、前記セル図形内の罫線の状況又は前記セル構成データより矩形に分割された矩形図形を示す情報であることを特徴とする請求項３記載の表ベクトルデータ生成方法。
【請求項５】
前記色塗りアウトラインは矩形図形の集合として表現されることを特徴とする請求項１記載の表ベクトルデータ生成方法。
【請求項６】
前記生成する工程は、前記判定する工程で前記表枠のセルを構成していると判定されなかった場合、前記変換する工程で生成された色塗りアウトラインとを組み合わせる一方で、前記変換する工程で生成された罫線は罫線としてデータ化することを特徴とする請求項１記載の表ベクトルデータ生成方法。
【請求項７】
画像データを入力する工程と、入力された画像データを２値化する工程と、２値化された２値データから前記表を構成する２値データを分離する工程とを更に有することを特徴とする請求項１記載の表ベクトル生成方法。
【請求項８】
表を構成する２値データから表ベクトルデータを生成する文書処理装置であって、
前記２値データをアウトライン化したデータが表枠のセルを構成しているか否かを判定する判定手段と、
前記判定手段により前記表枠のセルを構成していると判定されたデータを罫線と色塗り

アウトラインで表枠を表現するデータに変換する変換手段と、
前記変換されたデータと前記判定手段により前記表枠のセルを構成しないと判定されたデータとを合成し、表枠を表現するベクトルデータを生成する生成手段と、
を有することを特徴とする文書処理装置。
【請求項９】
請求項１記載の表ベクトルデータ生成方法をコンピュータに実行させるためのプログラム。
【請求項１０】
請求項９記載のプログラムを記録したコンピュータ読み取り可能な記録媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、スキャナなどの入力装置より読み込まれた紙文書を編集可能な電子データへ変換する技術に関し、特に紙文書中の表枠オブジェクトを解析し、罫線に置き換える技術に関する。
【背景技術】
【０００２】
近年、情報の電子化が進み、文書を紙ではなく電子化して保存或いは送信するシステムが急速に普及している。特に、フルカラーの文書を保存、送信に適した電子データとしては、紙原稿を文字、表、図等のオブジェクトへ像域分離し、各オブジェクトに適した形態でデータ化したベクトルデータが適しており、データ量を削減できるだけでなく再利用性が高い。
【０００３】
ここで、文字、表、線等のオブジェクトは、オブジェクトの外形をアウトライン化し、直線及び曲線により表現された形態へ変換することにより、各オブジェクトのデータ量を軽減できるだけでなく、文字は解像度に依存しない高画質な電子データへ、また表、線等の図形要素は、要素毎の編集が簡便な電子データへ変換することができる。
【０００４】
しかし、表枠はアウトライン化することで変倍しても画質劣化しない高画質電子データになるが、外側アウトラインと内側アウトラインからなるデータであるため、表の罫線毎には利用することができない。ここで、表構造を解析し表枠を罫線で置き換えれば、枠はアウトラインではなく太さを持つ罫線であるので再利用性が増し、また表構造も解析していることでセル毎の再利用も可能となる。また、外輪郭と内輪郭を一つの罫線へ置き換えることで、原本を表現するベクトルデータのデータ量として、データ量を削減することが可能となる。
【０００５】
このような表の再利用に関して、表構造の解析技術が種々提案されている。例えば黒画素の連結成分に着目し、表構造を解析する方法として、例えば特許文献１に記載のように、罫線が存在すると思われる小矩形の縦、横方向のヒストグラムを取り、ヒストグラムより線の位置、線種、線幅を認識する方法がある。また、表枠のアウトラインを一旦抽出して表構造を解析する手法として、例えば特許文献２があり、表枠アウトラインの外側輪郭と内側輪郭との構成より罫線を抽出している。他にも、例えば特許文献３記載のように、文字ブロックの構成より正確に罫線の構成を認識する手法がある。
【特許文献１】特開平４－１２３２８２号公報
【特許文献２】特開平５－１２４８９号公報
【特許文献３】特開平５－３３４４９０号公報
【特許文献４】特開平５－１０８８２３号公報
【発明の開示】
【発明が解決しようとする課題】
【０００６】

上記従来手法のように表構造、罫線を認識する手法は種々開示されているが、ベクトルデータとして罫線等で原本に忠実な表枠を表現しようとした場合に問題が生じる。

【０００７】

例えば、ヒストグラムによる黒画素の連結成分に着目する手法では、認識結果が誤っていた場合に原本を損なう危険がある。また、微妙な曲線で表現されるような表に対しての対処も不可能であり、無理に認識処理を施せば原本を損なってしまう。このような画質の劣化を防止するため、認識結果に基づき罫線化できない部分についてアウトライン化することも可能であるが、罫線化した部位とアウトライン化した部位との連結部について原本を損なうことなく表現することが難しい。

【０００８】

また、文字ブロックの構成により表構造を認識する従来手法では、表構造を忠実に認識可能であっても罫線等の位置、線幅を忠実に再現することができない。

【０００９】

また、表枠を一旦アウトライン化し、アウトラインの外側輪郭と内側輪郭との構成より認識する従来手法では、線の太さが一律であることを想定しているため、例えば各線の太さが極端に異なる表に対する場合、またノイズ等により罫線が途切れてしまっている原稿に対する場合に問題が生じる。これらの途切れ線については、再度元原稿について詳細に処理する必要がある、もしくは二値画像をより詳細に検証することで線を判定する、等のより詳細な検証処理により罫線認識結果を上げるしかない。しかし、認識結果の向上には技術的に限界がある。

【００１０】

本発明は、スキャン画像のようなノイズの多い原稿に対して、原本を損なうことなく、原本に忠実な表枠を表現するベクトルデータを作成することを目的とする。

【課題を解決するための手段】

【００１１】

本発明は、表を構成する２値データから表ベクトルデータを生成する表ベクトルデータ生成方法であって、前記２値データをアウトライン化したデータが表枠のセルを構成しているか否かを判定する工程と、前記判定する工程で前記表枠のセルを構成していると判定されたデータを罫線と色塗りアウトラインで表枠を表現するデータに変換する工程と、前記変換されたデータと前記判定する工程で前記表枠のセルを構成しないと判定されたデータとを合成し、表枠を表現するベクトルデータを生成する工程とを有することを特徴とする。

【００１２】

また、本発明は、表を構成する２値データから表ベクトルデータを生成する文書処理装置であって、前記２値データをアウトライン化したデータが表枠のセルを構成しているか否かを判定する判定手段と、前記判定手段により前記表枠のセルを構成していると判定されたデータを罫線と色塗りアウトラインで表枠を表現するデータに変換する変換手段と、前記変換されたデータと前記判定手段により前記表枠のセルを構成しないと判定されたデータとを合成し、表枠を表現するベクトルデータを生成する生成手段とを有することを特徴とする。

【発明の効果】

【００１３】

本発明によれば、原稿中の表を解析し、罫線が認識できる部位は罫線で表現し、罫線が認識できない部位についてはアウトラインで表現することで、複雑な形状をした表、特殊な形状をした表、スキャン画像のようなノイズの多い原稿中の表等に対し、強引な表解析による原稿の損失を防止しつつ、原稿中の表をベクトル表現することが可能である。尚、罫線が認識可能な部位については表構造が認識できているため、表ベクトルデータとして再利用が可能である。

【００１４】

また、表枠を外側輪郭及び内側輪郭のアウトラインで表現するより罫線で表現する方が

ベクトル表現としてデータサイズが小さくなるという効果がある。

【００１５】

更に、色塗りセルを作成する際に、セルを確認できない部位について色塗りセルを作成するので、極端に太い罫線等は色塗りセルを用いて自動的に表現されるために、表解析による誤認識においても可視的に同等な表構造を構築することが可能である。罫線とアウトラインによる表枠のベクトル表現は、表解析技術には、常に限界が存在するため、原稿の損失を防止しベクトル表示するには必須の手段である。

【発明を実施するための最良の形態】

【００１６】

以下、図面を参照しながら発明を実施するための最良の形態について詳細に説明する。

【実施例１】

【００１７】

図１は、実施例１における文書処理装置の外観を示す図である。図１において、１０１はコンピュータ装置であり、後述するフローチャートを参照して説明する処理を実現するためのプログラムを含む、文書の電子化処理プログラムを実行する。また、コンピュータ装置１０１は、ユーザに状況や画像を表示するためのディスプレイ装置１０２と、ユーザの操作を受け付けるキーボードやマウス等のポインティングデバイスを含んで構成される入力装置１０３とを付随する。このディスプレイ装置１０２としては、ＣＲＴやＬＣＤ等が用いられる。１０４はスキャナ装置であり、文書画像を光学的に読み取って電子化し、得られた画像データをコンピュータ装置１０１に送る。尚、スキャナ装置１０４としては、カラースキャナを用いるものとする。

【００１８】

図２は、実施例１における文書処理装置の構成の一例を示すブロック図である。図２において、２０１はＣＰＵであり、後述するＲＯＭ又はＲＡＭに格納された制御プログラムを実行することにより、後述する電子化処理を含む各種機能を実現する。２０２はＲＯＭであり、ＣＰＵ２０１によって実行される各種制御プログラムや制御データが格納されている。２０３はＲＡＭであり、ＣＰＵ２０１によって実行される各種制御プログラムを格納したり、ＣＰＵ２０１が各種処理を実行するのに必要な作業領域が定義されている。

【００１９】

２０４は外部記憶装置であり、詳細は後述する実施例１における処理をＣＰＵ１０１によって実現するための制御プログラムや、スキャナ装置１０４で読み取って得られた文書画像データ等を格納する。そして、２０５はコンピュータバスであり、上述した各構成を接続するものである。

【００２０】

図３は、文書処理装置における文書の電子化処理の概要を示す図である。ここで、電子化処理の流れは、まず入力部３０１において、電子化の対象であるカラー文書をスキャナ装置１０４によって読み込み、画像データとして外部記憶装置２０４に格納する。次に、２値化処理３０２において、後段の像域分離処理、アウトライン生成処理のために、外部記憶装置２０４に格納された文書の画像データに対して２値化処理を施す。そして、像域分離処理３０３では、２値化処理３０２で得られた２値画像から、文字、図、表、枠、線などの要素を抽出し、各領域に分割する。

【００２１】

次に、ベクトル化処理３０４において、領域分割された画像データに対して、文字部は文字認識部３０５で文字認識を行い、アウトライン作成部３０６でアウトラインベクトルデータへ変換する。また、表、枠の要素については、アウトライン作成部３０７でアウトラインデータ化し、表処理部３０８でアウトラインを罫線化する。尚、アウトライン作成部３０６及び３０７で変換された画像データは、各オブジェクトの輪郭線が滑らかな曲線により表現される高画質で、解像度に依存しない、かつ編集容易なベクトルデータへ変換される。

【００２２】

一方、その他の図、写真画、背景については、例えば背景については、圧縮部３０９でＪＰＥＧ圧縮など各々に適した形態で保持、圧縮する。

【００２３】

次に、電子文書作成処理３１０は、分割された要素毎の属性に基づいて文字認識データや表構造データを用い、それぞれ変換された画像データに基づき電子化文書を作成する。そして、出力部３１１は生成された電子化文書を外部記憶装置２０４に格納する。

【００２４】

尚、出力部３１１の出力形態は外部記憶装置２０４への格納に限られるものではなく、ディスプレイ装置１０２へ表示したり、不図示のネットワークインターフェースを介してネットワーク上の他の装置へ出力したり、不図示のプリンタへ送出したりすることも可能である。

【００２５】

ここで、図１及び図２に示す文書処理装置において実行される文書の電子化処理（図３参照）における各処理の詳細について、以下順に説明する。

【００２６】

［２値化処理］

２値化処理３０２では、入力された文書画像データより輝度情報を抽出し、その輝度値のヒストグラムを作成する。ヒストグラム上より複数の閾値を設定し、各々の閾値で２値化された２値画像上の黒画素の連結等を解析することで最適な閾値を導出し、その閾値による２値画像を得る。

【００２７】

［像域分離処理］

像域分離処理３０３とは、図４に示す左側の読み取られた１ページのイメージデータをオブジェクト毎の塊（ブロック）として認識し、各々の塊を文字／図画／写真／線／表等の属性に判定し、図４に示す右側のように、異なる属性（TEXT／PICTURE／PHOTE／LINE／TABLE）を持つ領域に分割する処理である。

【００２８】

像域分離処理３０３では、２値化処理３０２で得られた２値画像より、黒画素の輪郭線追跡を行って黒画素輪郭で囲まれる画素の塊を抽出する。また、面積の大きい黒画素の塊については、内部にある白画素に対しても輪郭線追跡を行い、白画素の塊を抽出し、更に一定面積以上の白画素の塊の内部からは再帰的に黒画素の塊を抽出する。

【００２９】

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

【００３０】

図５は、像域分離処理３０３で分離された各ブロックに対するブロック情報と入力ファイル情報を示す図である。図５に示すように、ブロック情報は、各ブロックの属性、座標（Ｘ，Ｙ）、幅（Ｗ）、高さ（Ｈ）、ＯＣＲ情報を含み、属性１は文字、属性２は図画、属性３は表、属性４は線、属性５は写真である。そして、入力ファイル情報は、ブロック総数Ｎ（図５に示す例では、ブロック１～ブロック６までの６である）を有する。

【００３１】

尚、各ブロックに対して、より鮮明な２値画像を得ようとした場合は、ここでブロック毎に上述した２値化処理を行っても良い。

【００３２】

［文字認識］

文字認識部３０５では、文字単位で切り出された画像に対して、パターンマッチングの

一手法を用いて認識を行い、対応する文字コードを得る。この認識処理は、文字画像から得られる特徴を数十次元の数値列に変換した観測特徴ベクトルと、予め字種毎に求められている辞書特徴ベクトルとを比較し、最も距離の近い字種を認識結果とする処理である。この特徴ベクトルの抽出には種々の公知手法があり、例えば文字をメッシュ状に分割し、各メッシュ内の文字線を方向別に線素としてカウントしたメッシュ数次元ベクトルを特徴とする方法がある。

【００３３】

像域分離処理３０３で抽出された文字領域に対して文字認識を行う場合、まず該当領域に対して横書き、縦書きの判定を行い、それぞれ対応する方向に行を切り出し、その後、文字を切り出して文字画像を得る。この横書き、縦書きの判定は、該当領域内で画素値に対する水平／垂直の射影を取り、水平射影の分散が大きい場合には横書き領域と判定し、垂直射影の分散が大きい場合には縦書き領域と判定すれば良い。また、文字列及び文字への分解は、横書きならば水平方向の射影を利用して行を切り出し、更に切り出された行に対する垂直方向の射影から、文字を切り出すことで行う。縦書きの文字領域に対しては、水平と垂直を逆にすれば良い。尚　この時、文字のサイズが検出できる。

【００３４】

［アウトライン生成部］

アウトライン作成部３０６、３０７では、像域分離処理で得られた文字、表、枠、線について、輪郭形状を直線及び滑らかな曲線により表現されるアウトラインベクトルデータに変換する。この手法は、オブジェクト原型よりアウトラインベクトルデータを作成する際に、画質劣化を抑えつつ、高速に処理する手法であり、詳細に説明する。

【００３５】

図６は、アウトライン作成部３０６、３０７の処理を示すフローチャートである。この処理の入力は、像域分離処理３０３で抽出された、例えば図４に示す文字（TEXT）領域の２値画像である。また、文字の場合は、文字認識部３０５で文字単位に切り出された画像であっても良い。

【００３６】

まず、ステップＳ６０１において、２値のラスター画像データを水平ベクトル及び垂直ベクトルからなるアウトラインデータ（以下、粗輪郭データと呼ぶ）へと変換する。尚、入力されるラスター画像データより抽出される粗輪郭データは一つだけとは限らず、殆どの場合、複数の粗輪郭データが抽出される。

【００３７】

次に、ステップＳ６０２において、抽出された粗輪郭データに対して、一粗輪郭データ毎に直線及び曲線により表現されるアウトラインベクトルデータへと変換する。

【００３８】

以下、図７及び図８を参照して図６に示すフローチャートの各ステップの処理について詳細に説明する。

【００３９】

ステップＳ６０１では、２値のラスター画像データを粗輪郭データへと変換する。図７は、ここで扱うラスター画像データの１画素を示す図である。図７に示すように、ラスター画像データにおける１画素は、４つの頂点を有し、垂直ベクトル及び水平ベクトルより構成される正方形として扱う。１画素を４つの頂点を有する正方形として扱い、その集合であるラスター画像データのアウトラインを抽出すると、得られるアウトラインデータは、水平ベクトル及び垂直ベクトルからなる粗輪郭データが抽出される。

【００４０】

このような粗輪郭データの抽出方法は、種々提案されており、特に特許文献４に開示されている粗輪郭抽出方法を用いれば、ラスター画像一面より効率良く、かつ高速に粗輪郭データを抽出することが可能である。

【００４１】

そして、抽出された輪郭データは、図８に示すような、水平ベクトル及び垂直ベクトル

が交互に並ぶ構成である粗輪郭データとなる。この粗輪郭データの抽出では、このような水平ベクトル及び垂直ベクトルが交互に並ぶ構成となる輪郭データを抽出し、次ステップＳ６０２へ進む。

【００４２】

図８は、粗輪郭データ及びアウトラインベクトルデータの一例を示す図である。図８において、（ａ）は粗輪郭データであり、（ｂ）はアウトラインベクトルデータである。

【００４３】

次に、ステップＳ６０２では、上述のステップＳ６０１で得られた粗輪郭データを直線及び曲線からなるアウトラインベクトルデータへと変換する。

【００４４】

図９は、実施例１における粗輪郭データをアウトラインベクトルデータへ変換する処理を示すフローチャートである。まず、粗輪郭データに対してノイズ除去を行い（ステップＳ９０１）、ノイズ除去された粗輪郭上の線分より主接線線分を抽出すると共に、準接線線分を抽出する（ステップＳ９０２）。尚、主接線線分、準接線線分については更に後述する。

【００４５】

次に、ステップＳ９０２で抽出された主接線線分、準接線線分よりアンカーポイントを抽出し（ステップＳ９０３）、抽出されたアンカーポイント間が数個の線分により構成されるグループを二次もしくは三次ベジェ曲線及び直線にあてはめる（ステップＳ９０４）。次に、残りの線分についてベジェ曲線近似を行い、三次もしくは二次のベジェ曲線により置き換える（ステップＳ９０５）。最後に、直線及び曲線より構成されるアウトラインベクトルデータに対して、補正処理を行う（ステップＳ９０６）。

【００４６】

以下、図１０乃至図１９を参照して図９に示したフローチャートの各ステップの処理について詳細に説明する。

【００４７】

［ノイズ除去］

まず、ノイズ除去（ステップＳ９０１）では、粗輪郭データよりノイズ除去を行う。図１０は、除去するノイズの一例を示す図である。尚、図中の“１”は、ラスター画像における１画素大のサイズを表し、１画素サイズの凹凸を除去することを目的とする。このノイズ除去では、図１０に示す（ａ）及び（ｂ）の網点ノイズ、同（ｃ）の角欠けノイズを除去するが、図１１に示すように、ノイズに似た粗輪郭データも存在する。特に、ここでは、小さな文字から大きな文字までを扱うことを前提としているので、図１１に示す形状のものを全て除去しては画質の劣化を招く。

【００４８】

よって、ノイズ解析が必要であり、例えば図１０に示すノイズは、それぞれ以下の条件（ａ）～（ｃ）を満たす場合に除去するものとする。

（ａ）１つの凸ノイズについて、次の式を満たす。

【００４９】

【数１】

$$\left|\vec{v}_1+\vec{v}_3\right| \geq \alpha_1 * \left|\vec{v}_2\right|$$

【００５０】

（ｂ）凸ノイズが複数個隣接している。

（ｃ）次式を全て満たす。

【００５１】

【数２】

$$\left|\vec{v}_1''\right| \geq \theta_1 、\left|\vec{v}_2''\right| \leq \theta_2 、\left|\vec{v}_3''\right| \leq \theta_3$$

【００５２】

尚、（ｂ）の除去手法としては、次の２つを比べ、小さい方側を凸ノイズの上辺としてノイズを除去する。

【００５３】

【数３】

$$\left|\vec{v}_1' + \vec{v}_3' + \ldots + \vec{v}_n'\right| \text{と} \left|\vec{v}_2' + \vec{v}_4' + \ldots + \vec{v}_{n-1}'\right|$$

【００５４】

ところで、ノイズを判断するための各パラメータα１，Θ_1，Θ_2，Θ_3は一定値でもよいが、小さなオブジェクトから大きなオブジェクトを扱う上で、全てのオブジェクトを一律に評価することは困難であるので、より詳細に行うためには、粗輪郭データそれぞれのオブジェクトサイズに応じて変更しても良い。このオブジェクトサイズの情報、即ち、文字サイズは文字認識部３０５により、またアウトラインサイズは像域分離処理３０３で既に抽出されているので、それらを用いて簡単に閾値Θ_1，Θ_2，Θ_3を導出することが可能である。

以上でノイズ除去が行えるが、元々粗輪郭抽出前に２値のラスター画像データにおいてノイズ除去することも可能であり、ラスター画像データでノイズ除去してあれば、ここで行わなくても良い。しかしながら、ラスター画像上でノイズを除去する場合は、画像一面を処理する必要があり、かつ上述した条件を満たす除去を行う場合は、非常に処理が重くなってしまう。これに対して、粗輪郭データでは扱うデータ量も少なくて済むので、非常に効率的である。

【００５５】

［接線線分抽出］

次に、ステップＳ９０２では、ノイズが除去された粗輪郭データより、オブジェクトに対する接線線分を抽出する。接線線分とは、粗輪郭データの線分中、ある線分がそのままオブジェクト形状の接線成分となる線分である。

【００５６】

図１２は、粗輪郭データより接線線分の抽出を説明するための図である。図１２に示す（ａ）は元の粗輪郭データであり、図１２に示す（ｂ）の太線部が粗輪郭（ａ）より抽出された接線線分である。ここで、接線線分は以下の条件（１）～（４）を満たす場合に、抽出される。

【００５７】

【数４】

① 図１２(a)のように、$\vec{a}_1 * \vec{a}_3 < 0$を満たすベクトル$\vec{a}_2$を構成する線分

② 図１２(b)のように、主接線線分に隣接し、線分の長さ$L1$が$L1 \geq \theta_4$を満たす線分 s1

③ 図１２(c)のように、線分の長さ$L2$が$L2 \geq \theta_5$を満たす線分 s2

④ 図１２(d)のように、隣接する線分の長さ$L3$,$L4$が
$(L2 \geq \theta_5 \,\&\&\, L2 \geq \theta_5) \,||\, (L2 \geq \theta_5 \,\&\&\, L2 \geq \theta_5)$を満たす線分 s3 及び s4

【００５８】

尚、条件に使用されるパラメータθ$_1$～θ$_5$は、解像度に依存する一定値でも構わないが、文字認識部３０５によって抽出される文字サイズ、像域分離処理３０３で検出される領域サイズ、ステップＳ６０１で検出されるアウトラインサイズ等のオブジェクトサイズにより、適応的に変更しても良い。

また、各オブジェクトサイズに応じて条件（１）～（４）のうち、適用する条件を選択しても良い。

【００５９】

オブジェクトのサイズにより条件を変更することで、文字サイズ、輪郭サイズに応じた最適な近似処理が可能となる。

【００６０】
そして、ステップＳ９０４～Ｓ９０６において、粗輪郭データを直線と曲線により表現されるアウトラインデータへと変換する。具体的には、曲線は図１３に示す（ａ）の三次ベジェ曲線と図１３に示す（ｂ）の二次ベジェ曲線を使用する。また図１３に示す（ｃ）は直線を示す。
【００６１】
尚、図１３に示す（ａ）の三次ベジェ曲線、図１３に示す（ｂ）の二次ベジェ曲線は、以下の式１、式２により表現される。
【００６２】
B（t）＝（1-t）3・Q1＋3（1-t）2・t・Q2＋3（1-t）・t2・Q3＋t3・Q4　…式１
B（t）＝（1-t）2・Q1'+2（1-t）・t・Q2'＋t2・Q3'　…式２
図１３において、点Ｑ１，Ｑ４，Ｑ１’，Ｑ３’，Ｑ１”，Ｑ２”をアンカーポイントとし、曲線を制御しているＱ２，Ｑ３，Ｑ２’をコントロールポイントと呼ぶ。ここで、コントロールポイントとアンカーポイントを結ぶ直線、例えば直線Ｑ１Ｑ２は、アンカーポイントＱ１において曲線と接する。
【００６３】
また、アンカーポイント間にコントロールポイントがなければ、図１３に示す（ｃ）のように直線となる。
【００６４】
［アンカーポイント抽出］
ステップＳ９０３では、上述のステップＳ９０２で抽出された接線線分上に新たな点を抽出し、それをアンカーポイントとする。このアンカーポイントは、接線線分の端２つに対してそれぞれ抽出される。よって、一つの接線線分に対して２つのアンカーポイントが抽出されるが、２つのアンカーポイントが一致した場合には一つのアンカーポイントのみ抽出されるものとする。２つのアンカーポイントが抽出される場合は、アンカーポイントに挟まれた部位は自動的にオブジェクト上の直線となる。
【００６５】
ここで、接線線分上の一つの端点に対するアンカーポイントの抽出方法の一例について説明する。図１４は、アンカーポイントの抽出方法の一例を示す図である。図１４に示すＶ２を接線線分のベクトルとし、ベクトルＶ１側の端点に対するアンカーポイントの抽出方法について説明する。
【００６６】
まず、ベクトルＶ２に隣接するベクトルＶ１が接線線分であれば、その端点をアンカーポイントとする。隣接する線分が接線線分でない場合は、図１４に示す（ａ）のように、ベクトルＶ２上端点よりa|V1|となる点をアンカーポイントとする。図１４に示す（ｂ）のように|V2|/2＜a|V1|となる場合は、Ｖ２ベクトルの中心点をアンカーポイントとする。
【００６７】
［一次近似、二次近似］
次に、ステップＳ９０４、Ｓ９０５では、上述のステップＳ９０３で抽出されたアンカーポイント間をベジェ関数で曲線近似する。尚、ステップＳ９０３で自動的に直線属性となった線分に対しては曲線近似処理を行わない。
【００６８】
曲線近似処理は、具体的には２つの種類の近似処理からなる。まず、アンカーポイントの間が数個（＜ｎ１）の線分から構成されるようなオブジェクト上の細かい部位を纏めて一つの曲線で置き換える一次近似処理（ステップＳ９０４）と、数個より多い線分から構成される線分に対して１つ或いは複数の曲線を用いて近似する二次近似処理（ステップＳ９０５）とである。
【００６９】
前者の手法は、線分の組み合わせに対して１つの曲線を当てはめる処理であるが、後者

の手法を用いても数個の線分に対して近似を行うことも可能なため、後者の手法のみを用いてアンカーポイント間を曲線近似処理しても良い。しかしながら、前者の手法は、後者の手法に比べ、パフォーマンスの点で優れており、また少ない線分の組み合わせに対して確実に少ないポイント数で近似できるため、細かい部位については一次近似を用いることが望ましい。

【００７０】

まず、図１５を参照して一次近似処理（ステップＳ９０４）の一例について説明する。ここで図１５に示す点Ａ１、Ａ２がそれぞれステップＳ９０３で抽出されたアンカーポイントとする。そして、アンカーポイント間の線分Ｌ０，Ｌ１，Ｌ２に対して、Ｃ１，Ｃ２といったコントロールポイントを設けることで曲線を近似する。

【００７１】

尚、Ｃ１、Ｃ２の値はＬ０，Ｌ２との関係から求められる。また、アンカーポイント間が数個の線分により構成され、両端のアンカーポイントに対する接線成分が直交している場合は二次ベジェ曲線で置き換える。また、数個の線分がオブジェクトの大きさに対して十分大きければ、三次ベジェを用いてより精密に置き換えても良い。

【００７２】

ここで、一次近似処理はパターンに応じた置き換えであり、ステップＳ９０３のアンカーポイントの抽出もパターンに応じた処理であるため、これら２つのステップをまとめて行っても良い。

【００７３】

次に、二次近似処理（ステップＳ９０５）について説明する。まず、二次近似処理で使用する曲線を図１６に示す。図１６に示すように、曲線は三次ベジェ曲線であり、アンカーポイントＰ０，Ｐ３を結ぶ直線と、コントロールポイントＰ１，Ｐ２を結ぶ直線とは平行になるよう構成されている。このような平行制限を設けると、三次ベジェ曲線Ｌ０上の点で直線Ｐ０Ｐ３より最も離れた点Ｐｆとの距離をＤｆ、直線Ｐ０Ｐ３とコントロールポイントＰ１，Ｐ２との距離をＤｃとすると、次式の関係が成り立つ。

【００７４】

$$Dc = 4/3Df \quad \cdots式3$$

尚、平行制限を用いたベジェ曲線を使用することで、近似処理を簡易に行うことが可能となる。

【００７５】

以下、近似処理の概要について説明する。二次近似処理では、まず区分曲線に分割し、各区分曲線に対して曲線近似処理を行う。ここで、区分曲線とは、図１６に示すように、曲線が１つの弧を描く、即ち三次曲線において２つのアンカーポイントによる直線に対して２つのコントロールポイントが同方向に構成されているような曲線である。

【００７６】

区分曲線への分割では、まず図１７に示す（ｂ）のように、複数の線分の組み合わせにより、パターンマッチング的に方向ベクトルを抽出する。求められた方向ベクトルの変化を追っていき、方向ベクトル変化の正負が変化した点が分割点である。

【００７７】

尚、上述の分割点は、曲線近似におけるアンカーポイントとなり、アンカーポイントにおける接線ベクトルは、方向ベクトルがそのままなる。

【００７８】

また、図１７に示す（ａ）は、区分曲線へ分割した例を示す図である。

【００７９】

次に、図１８を参照して区分曲線に対する曲線近似処理について説明する。図１８では、一つの区分曲線を示しており、区分曲線上の線分郡よりＮ個の点を抽出したものをそれぞれｐ１，ｐ２，…，ｐＮとする。このとき、区分曲線の始点ｐ１、終点ｐＮはアンカーポイントである。

【００８０】

尚、各アンカーポイントにおける接線線分は、ステップＳ９０５もしくは区分曲線への分割におけるアンカーポイント抽出時にそれぞれ抽出されている。

【００８１】

ここで、アンカーポイントｐ１、ｐＮを結ぶ線分ｐ１ｐＮより最も距離の離れている曲線上の点ｐｆを求める。二次近似処理においては、関数近似処理を簡易に行うため、コントロールポイントを結ぶ線分Ｃ１Ｃ２が線分ｐ１ｐＮに対して平行となるように近似する。よって、点ｐｆと線分ｐ１ｐＮとの距離をＬとすると、点Ｃ１、Ｃ２より線分ｐ１ｐＮへの距離が（４／３）×Ｌとなるように、Ｃ１、Ｃ２を求める。

【００８２】

例えば、ｐｆの座標値が（ｐfx，ｐfy）であった場合、ｐ１、ｐＮの各座標値（ｐ1x，ｐ1y）、（ｐNx，ｐNy）とｐ１における接線ベクトルｐ１Ｃ１（ｐcx，ｐcy）を用いると、Ｃ１の座標値（Ｃ1x，Ｃ1y）は、

$$C1x = K \times pfx + p1x$$
$$C1y = K \times pfy + p1y$$
$$K = (3p1x - 4pfx)(pNy - p1y) + (pNx - p1x)(4pfy - 3p1y) + p1x(pNy - p1y) - p1y(pNx - p1x) / (3(pNy - p1y)pcx + 3(pNx - p1x)pcy)$$

となり、ｐｆの座標値より一意に決定することができる。また、Ｃ２についても、Ｃ１と同様に求めることが可能である。

【００８３】

以上の区分曲線への曲線近似処理を全てのオブジェクト上全ての区分曲線へ行うことで、オブジェクトのアウトラインは直線とベジェ曲線により構成されるアウトラインデータへと変換される。

【００８４】

［補正処理］

以上、ステップＳ９０１～Ｓ９０５により、オブジェクトの外形を直線及び曲線により構成されたアウトラインベクトルデータへ変換できるが、本手法では水平ベクトルと垂直ベクトルのみを使用した粗輪郭データから変換するために、また処理を効率化して行っているために、一連のステップで作成されたアウトラインベクトルデータは一種の癖をもつベクトルデータとなる。そこで、ステップＳ９０６では、アウトラインベクトルデータを解析し、これらの癖を補正する処理を行う。

【００８５】

図１９は、具体的にアウトラインベクトルデータの癖を表した図である。水平ベクトルと垂直ベクトルのみの粗輪郭データを用いて解析し、変換しているため、原図形における斜め直線は、曲線により表現されている。これらについては、アンカーポイント間を結ぶ直線とコントロールポイントとの距離を調べ、斜め直線か否かを判定する。ここで、斜め直線と判定された場合、アンカーポイント間のコントロールポイントを排除して斜め直線に置き換える。

【００８６】

［表処理部］

次に、実施例１における表処理部３０８について説明する。表処理部３０８では、表中のセル及びその構成を認識し、表枠を罫線によって表現する等、セル毎に編集可能な電子データへ変換する。尚、表部は、像域分離処理３０３により表枠として表枠中の文字部と分離して抽出されているものとする。また、表処理部３０８では、表中の文字部も含めて処理可能である。

【００８７】

図２０は、実施例１における表処理を示すフローチャートである。尚、入力データは、アウトライン作成部３０７によりアウトライン化されたデータである。

【００８８】

まず、アウトライン作成部３０７から入力されたアウトラインデータに対して、アウト

ライン毎にセル認識を行い、セルであるアウトラインと、セルでないアウトラインに分離する（ステップＳ２００１）。ここで、セルであると識別されたアウトラインについては後述するセル変換を行い、四点で記述されるセルへ変換する（ステップＳ２００２）。

【００８９】

次に、四点で記述されたセルを用いてセルの構成を認識し、罫線と色塗りアウトラインとで表現される表に変換する（ステップＳ２００３）。ここで、色塗りアウトラインについては、上述のステップＳ２００１でセルでないと判断されたアウトラインと組み合わせてアウトラインの表枠を生成する（ステップＳ２００４）。最後に、ステップＳ２００３で生成された罫線と、ステップＳ２００４で生成された罫線とを用いて表枠データを生成する（ステップＳ２００５）。

【００９０】

ここで、図面を参照して図２０に示すフローチャートの各ステップの処理について詳細に説明する。

【００９１】

［セル構成アウトライン判定］

ステップＳ２００１では、アウトライン作成部３０７でアウトライン化されたデータを用いて、そのアウトラインがセルを構成しているアウトラインであるか否かを判定する。ここで、元々アウトラインは外輪郭と内輪郭に分類されているが、外輪郭のうち、表全体の外枠を構成しているアウトラインを抽出する。尚、表の内部に表が存在するような場合もあるので、外枠は複数抽出される場合もある。

【００９２】

次に、外枠の内側に存在するセルを構成しているアウトラインを抽出する。尚、ここでの処理はアウトラインよりそのアウトラインがセルを構成するサイズであるか否かを判定し、更にアウトラインを図形認識処理し、アウトラインが矩形図形、もしくは三角図形を構成しているか否かを判定する。また、矩形図形、三角図形、もしくは矩形図形の集合と判定されたアウトラインをセルアウトラインとする。

【００９３】

図２１は、矩形図形、三角図形、矩形図形の集合と判定されるアウトラインの一例を示す図である。

【００９４】

［セル図形変換］

ステップＳ２００２では、ステップＳ２００１で外枠、もしくはセルを構成していると判定されたアウトラインをセル図形へ変換する。まず、ステップＳ２００１で外枠を構成するアウトラインと内部セルを構成するアウトラインが抽出されているが、外枠を構成するアウトラインの角度が全て90°で表現される図形であると判定された場合、90°角の間を直線で表現した図形へ変換する。次に、外枠の内側のセルを構成すると判定されたセルについてセル図形へ変換する。ここで、セル図形とは矩形図形である。

【００９５】

例えば、図２１に示すセルアウトラインをセル図形へ変換した例を図２２示す。図２１に示すセルアウトライン（ａ）～（ｃ）はそれぞれ図２２に示す（ａ）～（ｃ）のように変換される。図２１に示す（ａ）のセルアウトラインは、矩形図形の当てはめ処理によりそのまま図２２に示す（ａ）となる。また、図２１に示す（ｂ）のような三角図形のセルアウトラインに対しても同様に、矩形図形の当てはめ処理を行う。この三角図形に対する矩形図形の当てはめ処理では、三角を構成するセルアウトラインを囲むようにセル矩形を当てはめる。

【００９６】

尚、当てはめられた矩形図形は最終的に、その位置関係よりセル図形同士が統合され、一つのセル図形として抽出される。例えば、図２２に示す（ｂ）のセル図形は、図２３に示すセル図形と統合され、一つのセル図形として表現される。統合されないセル図形も当然あり、その図形についてはそのまま三角アウトラインに当てはめられたセル図形をセル

図形とする。

【００９７】

また、図２１に示す（ｃ）のような矩形図形の集合として抽出されるようなセルアウトラインは、図２２に示す（ｃ）のように、それぞれの矩形図形へ分離する。この矩形図形への分離処理では、アウトラインの中の直角をなすであろう角を検出し、その角点の構成から矩形図形へ分解する。

【００９８】

このように、外枠図形とセル図形とを抽出し、各セル図形はその構成されるセルアウトラインにより属性情報が付加される。

【００９９】

図２４は、セル図形を構成するセルアウトラインと各属性情報についての一例を示す図である。

【０１００】

［セル構成認識］

ステップＳ２００３では、ステップＳ２００２で変換されたセル図形を用いて、各セル図形のセル構成を認識する。

【０１０１】

図２５は、セル構成を認識する認識処理を示すフローチャートである。まず、抽出されたセル全てを用いて表の水平方向及び垂直方向を求め、セル図形の全てを求められた水平方向及び垂直方向の成分からなるセル図形へ変換する（ステップＳ２５０１）。次に、外枠を用いてセル図形をマッピングするマッピング領域を作成し（ステップＳ２５０２）、外枠内にあるセル図形全ての結合関係を調べ、領域上にマッピングする（ステップＳ２５０３）。このとき、外枠が複数存在するときは、外枠サイズの小さいもの、即ち重なっている場合には内部の表から順に行う。最後に、セルの結合関係より罫線、及び色塗りセルを導き出し（ステップＳ２５０３）、太さ、罫線の位置調整を行い（ステップＳ２５０５）、表枠を出力する。

【０１０２】

ここで、図面を参照して図２５に示すフローチャートの詳細について説明する。

【０１０３】

ステップＳ２５０１では、セル図形を整える。まず、全てのセルの角度と長さの関係より表枠の水平方向及び垂直方向の平均となる方向ベクトルを抽出する。ここで抽出された垂直方向のベクトル及び水平方向のベクトルをそれぞれｖ，ｈとする。次に、全てのセル図形を今求めたベクトルｖ，ｈで構成されるセル図形へ変換する。このとき、外枠も求められた水平ベクトルｈと垂直ベクトルｖから構成されるよう変換する。

【０１０４】

ステップＳ２５０２では、ステップＳ２５０１で抽出した外枠を用いてマッピング領域を作成する。マッピング領域とは、セル図形をマッピングするための領域であり、外枠内部の領域がそのままマッピング領域となる。ここで、外枠の交点を抽出しておく。交点とは、表の罫線と罫線が交差する点のことであり、外枠においては、外枠の角点がそのまま交点となる。図２６にマッピング領域と交点の一例を示す。

【０１０５】

ステップＳ２５０３では、表構成を認識する。表構成の認識では、ステップＳ２５０２で抽出されたマッピング領域内にセル図形をマッピングしていき、マッピングされるセル図形より表の罫線と罫線が交差する交点を抽出していくことで表構成を認識する。即ち、交点の隣接関係を調べていくことで、表構成を認識する。

【０１０６】

図２７は、ステップＳ２５０３の表構成を認識する処理の詳細を示すフローチャートである。まず、ステップＳ２７０１において、抽出されている交点より注目点を抽出する。ここで注目点とは、右側と下側に隣接し、繋がっている交点を持つ交点であり、かつその３点を含む矩形領域がマッピング領域であり、かつその矩形領域に対してまだ何もマッピ

ングされていない交点である。図２８は、マッピング領域と交点とマッピングセルと注目点の関係を示す図である。

【０１０７】

次に、ステップＳ２７０２において、注目点に対して、注目点と左上の角点が一致するセル図形が存在するか否かを判定する。具体的には、未だマッピングされていない全てのセル図形の左上の角点と注目点との距離を調べ、セル図形の左上の角点と注目点との距離が一定値以内で、最も注目点に近いセル図形を注目点と左上の角点が一致するセル図形とする。ここで、注目点と左上の角点が一致するセル図形が存在すればステップＳ２７０３へ進み、そのセル図形をマッピング領域上にマッピングする。また、セル図形全ての左上の角点と注目点との距離が一定値以内にあるセル図形が存在しない場合はステップＳ２７０４へ進み、注目点を左上にもつような色塗りセルを作成してマッピングする。

【０１０８】

この色塗りセルは、矩形図形である。まず、注目点とその隣接する右側と下側の交点より少し広げた矩形領域（以後矩形領域Ａと呼ぶ）内に、まだマッピングされていないセル図形の角点がないか判定し、もし角点が存在すればその角点を通る水平方向及び垂直方向の直線によって領域を区切る。この区切り作業を矩形領域Ａ内に存在する角点全てに対して行い、水平線及び垂直線によって区切られた領域の最も左上にある区切られた矩形図形を色塗りセルとし、マッピングする。図２９は、矩形領域Ａ内に色塗りセルを作成する例を示す図である。

【０１０９】

次に、ステップＳ２７０５において、ステップＳ２７０３、Ｓ２７０４でマッピングされたセル図形及び色塗り図形を用いて交点を作成する。この交点はこのマッピング図形の角点がそのまま交点となるが、もしマッピング図形の角点が、既に存在する交点との距離がある閾値以内であれば、その角点により作成される交点は既に存在すると判断できるため、その角点より新たな交点は作成しない。ここで、マッピング図形の左上の交点は注目点と一致と判断されているため、左上の角点より新たな交点は作成されない。また、マッピング図形の右上の角点より作成される交点は注目点より水平線上にあるとし作成し、左下の角点は注目点より垂直線上にあるとし作成する。

【０１１０】

そして、ステップＳ２７０６において、現在抽出されている交点の中で注目点があるか否かを判定する。注目点とは、上述したように、右側と下側に隣接し繋がっている交点を持つ交点であり、かつその３点を含む矩形領域がマッピング領域であり、かつ該矩形領域に対しまだ何もマッピングされていない交点である。交点が囲む領域内にセル図形及び塗りつぶしセルがマッピングされていない領域が存在すれば注目点は存在する。

【０１１１】

そして、注目点が存在しない場合には表構成認識処理を終了する。また、まだ注目点が存在する場合はステップＳ２７０１に戻り、再度注目点を抽出する一連の処理を繰り返す。

【０１１２】

以上の繰り返し処理により、交点の隣接関係が作成され、罫線を表現することが可能となる。図３０に作成された交点の隣接関係と、マッピングされたセル図形及び色塗りセル、またセル図形の場合はその属性情報を記述した例を示す。

【０１１３】

ここで、図２５に戻り、ステップＳ２５０４では、表中の罫線を抽出する。罫線の抽出にあたり、まず隣接する色塗りセルを結合する。また、例えば図２４に示す属性８に相当するセルについて結合処理を行う。この時、結合するセル間の交点は消去する。

【０１１４】

以上の処理を行った結果、図３０に示す交点の隣接関係は図３１のように変換される。結果として交点の水平方向及び垂直方向の繋がりは罫線として抽出することが可能である。また、例えば図２４に示す属性２、５、９等のセル図形がある場合は、各属性に応じて

斜線を追加する。尚、属性３，４，６，７等のセル図形については、各属性に応じて三角形色塗り領域を作成する。

【０１１５】

次に、ステップＳ２５０５では、罫線の太さ及び位置関係を調節する。罫線の太さは、ステップＳ２５０３でマッピングされたセル図形のうち、隣接するセル図形の距離から求められる。また、罫線の位置は隣接するセル図形の中間となるように調節する。

【０１１６】

以上、セルアウトラインとして抽出されたアウトラインの集合を、罫線及び、直線表現された色塗りセルにより構成されるベクトルデータへ変換される。

【０１１７】

［アウトライン表枠生成］

ステップＳ２００４では、ステップＳ２００３により作成された色塗りアウトラインとステップＳ２００１でセルアウトラインと判断されなかったアウトラインとについて合成処理し、可視的に表枠を表現するアウトライン表枠を生成する。図３２に示す（ａ）は、ステップＳ２００１によりセルアウトラインと判断されなかったアウトライン、同（ｂ）は、ステップＳ２００３により作成された色塗りアウトライン、同（ｃ）は、ステップＳ２００３により作成された罫線の例をそれぞれ示す図である。

【０１１８】

図３２に示す（ａ）において、３２０１～３２０３はノイズによりセルアウトラインと判断されなかったアウトラインの内側輪郭のアウトラインであり、３２０４～３２０７はノイズである。

【０１１９】

尚、３２０４～３２０７のノイズは３２０７のように、表枠を構成するようなノイズである場合もあり重要である。図３２に示す（ａ）と（ｂ）を組み合わせることで、図３３のようにアウトラインによる表枠を生成する。このとき、３２０１～３２０３は元々内側輪郭のアウトラインであり、即ち色抜きアウトラインである。一方、図３２に示す（ｂ）のアウトラインは色塗りのアウトラインである。これら色塗りと色抜きのアウトラインを組み合わせることで、表枠線を表現する。

【０１２０】

［表枠データ生成］

最後に、ステップＳ２００５において、ステップＳ２００４で生成されたアウトライン表枠とステップＳ２００３で生成された罫線とを用いて表枠を形成する。

【０１２１】

図３４は、図３３に示すアウトライン表枠と図３２に示す（ｃ）のステップＳ２００３で生成された罫線とを用いて表枠を構成する例を示す図である。

【０１２２】

尚、ここでの罫線は、セル一つ一つを矩形として表現する図３５に示す（ａ）のような罫線であっても、図３５に示す（ｂ）のような通常の罫線であっても構わない。

【０１２３】

以上のステップＳ２００１～Ｓ２００４のステップにより、表枠アウトラインは可視的に元原稿の状態を維持し、罫線に置き換えることが可能である。罫線と認識されるものについては罫線化する一方で、原稿上のノイズにより線が途切れてしまっている場合については、色塗りセルと元のアウトラインを使用することで罫線を表現する。また、表中に極端に太さが異なる線が存在しても、ステップＳ２００３により太線は色塗りセルとして表現されるため、可視的に元原稿と同等となる。

【０１２４】

［アプリデータへの変換処理］

以上の通り、１頁分のイメージデータを像域分離処理３０３し、ベクトル化処理３０４した結果は図３６に示すような中間データ形式のファイルとして変換される。このようなデータ形式は、ドキュメント・アナリシス・アウトプット・フォーマット（ＤＡＯＦ）と

呼ばれる。

【０１２５】

図３６は、ＤＡＯＦのデータ構造を示す図である。図３６において、３６０１はHeaderであり、処理対象の文書画像データに関する情報が保持される。３６０２はレイアウト記述データ部であり、文書画像データ中の文字（TEXT）、タイトル（TITLE）、キャプション（CAPTION）、線画（LINEART）、自然画（PICTURE）、枠（FRAME）、表（TABLE）等の属性毎に認識された各ブロックの属性情報とその矩形アドレス情報を保持する。３６０３は文字認識記述データ部であり、TEXT、TITLE、CAPTION等のTEXTブロックを文字認識して得られる文字認識結果を保持する。３６０４は表記述データ部であり、TABLEブロックの構造の詳細を格納する。３６０５は画像記述データ部であり、PICTUREやLINEART等のブロックのイメージデータを文書画像データから切り出して保持する。

【０１２６】

このようなＤＡＯＦは中間データとしてのみならず、それ自体がファイル化されて保存される場合もあるが、このファイルの状態では、所謂一般の文書作成アプリケーションで個々のオブジェクトを再利用することはできない。

【０１２７】

そこで、このＤＡＯＦからアプリケーションデータに変換する電子文書作成処理３０９について説明する。

【０１２８】

図３７は、電子文書作成処理の全体の概略を示すフローチャートである。まずステップＳ３７０１において、ＤＡＯＦデータの入力を行う。次に、ステップＳ３７０２において、アプリデータの元となる文書構造ツリー生成を行う。そして、ステップＳ３７０３で、文書構造ツリーに基づいてＤＡＯＦ内の実データを流し込み、実際のアプリデータを生成する。

【０１２９】

図３８は、文書構造ツリー生成処理の詳細を示すフローチャートである。また、図３９は文書構造ツリーを説明するための図である。尚、全体制御の基本ルールとして、処理の流れはミクロブロック（単一ブロック）からマクロブロック（ブロックの集合体）へ移行する。尚、以下の説明で、「ブロック」はミクロブロック及びマクロブロック全体を指すものとする。

【０１３０】

まず、ステップＳ３８０１では、ブロック単位に縦方向の関連性に基づいて再グループ化する。スタート直後はミクロブロック単位での判定となる。ここで、関連性とは、距離が近い、ブロック幅（横方向の場合は高さ）がほぼ同一であることなどで定義することができる。また、距離、幅、高さなどの情報はＤＡＯＦを参照し、抽出する。

【０１３１】

図３９は、ページの構成とその文書構造のツリーを示す図である。図３９に示す（ａ）は実際のページ構成、図３９に示す（ｂ）はその文書構造ツリーである。

【０１３２】

ステップＳ３８０１での結果、図３９に示すＴ３、Ｔ４、Ｔ５が１つのグループＶ１として生成され、Ｔ６、Ｔ７が１つのグループＶ２として生成され、図３９に示す（ｂ）のように、グループＶ１とグループＶ２が同じ階層のグループとして生成される。そして、ステップＳ３８０２において、縦方向のセパレータの有無をチェックする。セパレータは、例えば物理的にはＤＡＯＦ中でライン属性を持つオブジェクトである。また、論理的な意味としては、アプリ中で明示的にブロックを分割する要素である。ここでセパレータを検出した場合は、同じ階層で再分割する。

【０１３３】

次に、ステップＳ３８０３において、分割がこれ以上存在し得ないか否かをグループ長を利用して判定する。ここで、縦方向のグループ長がページ高さとなっている場合、文書構造ツリー生成を終了する。また、図３９に示す例の場合、セパレータもなく、グループ

高さはページ高さではないのでステップＳ３８０４へ進み、ブロック単位で横方向の関連性に基づいて再グループ化する。ここもスタート直後の第一回目はミクロブロック単位で判定を行うことになる。尚、関連性、及びその判定情報の定義は、縦方向の場合と同じである。

【０１３４】

図３９に示す例の場合、Ｔ１、Ｔ２でＨ１が、Ｖ１、Ｖ２でＨ２がＶ１、Ｖ２の１つ上の同じ階層のグループとして生成される。そして、ステップＳ３８０５において、横方向セパレータの有無をチェックする。図３９に示す例では、Ｓ１があるので、これをツリーに登録し、Ｈ１、Ｓ１、Ｈ２という階層を生成する。

【０１３５】

次に、ステップＳ３８０６において、分割がこれ以上存在し得ないか否かをグループ長を利用して判定する。ここで、横方向のグループ長がページ幅となっている場合、文書構造ツリー生成を終了する。また、そうでない場合はステップＳ３８０１に戻り、再びもう一段上の階層で、縦方向の関連性チェックから繰り返す。図３９に示す例の場合、分割幅がページ幅になっているので、ここで終了し、最後にページ全体を表す最上位階層のＶ０が文書構造ツリーに付加される。

【０１３６】

文書構造ツリーが完成した後、その情報に基づいてアプリデータを生成する（ステップＳ３７０３）。図３９に示す例の場合、具体的には、以下のようになる。

【０１３７】

即ち、Ｈ１は横方向に２つのブロックＴ１及びＴ２があるので、２カラムとし、Ｔ１の内部情報（ＤＡＯＦを参照、文字認識結果の文章、画像など）を出力後、カラムを変え、Ｔ２の内部情報出力、その後Ｓ１を出力する。次に、Ｈ２は横方向に２つのブロックＶ１及びＶ２があるので、２カラムとして出力、Ｖ１はＴ３、Ｔ４、Ｔ５の順にその内部情報を出力、その後カラムを変え、Ｖ２のＴ６、Ｔ７の内部情報を出力する。

【０１３８】

以上の処理によりアプリデータへの変換処理を行うことができる。

（変形例）

実施例１では、像域分離した表枠画像に対して表処理部３０８が表処理を行っているが、図２０に示す処理を表領域でない全ての画像領域に対して行うことも可能である。その場合、ステップＳ２００１のセルアウトラインの判定により、通常の文字等のアウトラインはステップＳ２００４へ進み、表構成アウトラインはステップＳ２００２へ進む。これ以降の処理は、実施例１で説明した通りであり、文字アウトラインと表枠アウトラインはステップＳ２００４で生成され、ステップＳ２００３で生成された罫線と合成することで、原稿上の文字、表、線画等をベクトル表現したベクトルデータを作成することが可能である。

【０１３９】

尚、本発明は複数の機器（例えば、ホストコンピュータ，インターフェース機器，リーダ，プリンタなど）から構成されるシステムに適用しても、１つの機器からなる装置（例えば、複写機，ファクシミリ装置など）に適用しても良い。具体的には、複合機や、複写機や、ファクシミリ装置で、高品位に変倍するために、スキャンした画像データを入力し（公衆回線やネットワークから画像データを入力しても良い）、画像データから輪郭ベクトルを抽出し、抽出した輪郭ベクトルを変倍し、変倍された輪郭ベクトルから画像データを生成し、生成した画像データをプリントする際の輪郭ベクトル抽出時に適用できる。

【０１４０】

また、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を記録した記録媒体を、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（ＣＰＵ若しくはＭＰＵ）が記録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、達成されることは言うまでもない。

【０１４１】

この場合、記録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録媒体は本発明を構成することになる。

【０１４２】

このプログラムコードを供給するための記録媒体としては、例えばフロッピー（登録商標）ディスク，ハードディスク，光ディスク，光磁気ディスク，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。

【０１４３】

また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、その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又は全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【０１４４】

更に、記録媒体から読出されたプログラムコードが、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込まれた後、そのプログラムコードの指示に基づき、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い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【図面の簡単な説明】

【０１４５】

【図１】実施例１における文書処理装置の外観を示す図である。

【図２】実施例１における文書処理装置の構成の一例を示すブロック図である。

【図３】文書処理装置における文書の電子化処理の概要を示す図である。

【図４】実施例１における像域分離処理を説明するための図である。

【図５】像域分離処理３０３で分離された各ブロックに対するブロック情報と入力ファイル情報を示す図である。

【図６】アウトライン作成部３０６、３０７の処理を示すフローチャートである。

【図７】ラスター画像データの１画素を示す図である。

【図８】粗輪郭データ及びアウトラインベクトルデータの一例を示す図である。

【図９】実施例１における粗輪郭データをアウトラインベクトルデータへ変換する処理を示すフローチャートである。

【図１０】除去するノイズの一例を示す図である。

【図１１】ノイズに似た粗輪郭データの一例を示す図である。

【図１２】粗輪郭データより接線線分の抽出を説明するための図である。

【図１３】粗輪郭データをアウトラインデータへと変換する際に使用される三次ベジェ曲線、二次ベジェ曲線、直線を示す図である。

【図１４】アンカーポイントの抽出方法の一例を示す図である。

【図１５】一次近似処理の一例を説明するための図である。

【図１６】二次近似処理で使用する曲線を示す図である。

【図１７】区分曲線へ分割した例とパターンマッチング的に方向ベクトルを抽出する例を示す図である。

【図１８】区分曲線に対する曲線近似処理を説明するための図である。

【図１９】具体的にアウトラインベクトルデータの癖を表した図である。

【図２０】実施例１における表処理を示すフローチャートである。

【図２１】矩形図形、三角図形、矩形図形の集合と判定されるアウトラインの一例を示す図である。

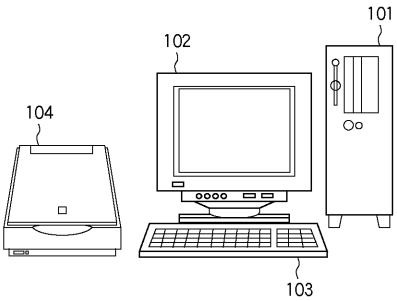
【図２２】図２１に示すセルアウトラインをセル図形へ変換した例を示す図である。

【図２３】隣接するセル図形の統合例を示す図である。

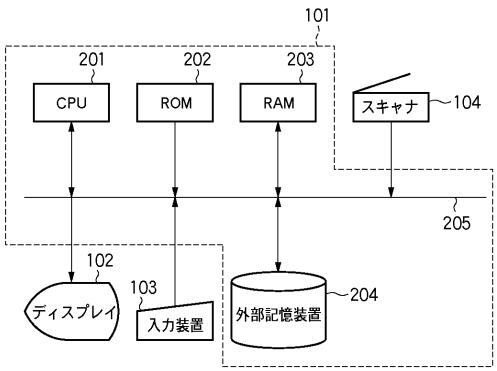
【図２４】セル図形を構成するセルアウトラインと各属性情報についての一例を示す図で

ある。
【図２５】セル構成を認識する認識処理を示すフローチャートである。
【図２６】マッピング領域と交点の一例を示す図である。
【図２７】ステップＳ２５０３の表構成を認識する処理の詳細を示すフローチャートである。
【図２８】マッピング領域と交点とマッピングセルと注目点の関係を示す図である。
【図２９】矩形領域Ａ内に色塗りセルを作成する例を示す図である。
【図３０】作成された交点の隣接関係と、マッピングされたセル図形及び色塗りセル、またセル図形の場合はその属性情報を記述した例を示す図である。
【図３１】図３０に示す交点の隣接関係から得られた処理結果を示す図である。
【図３２】（ａ）はステップＳ２００１でセルアウトラインと判断されなかったアウトライン、（ｂ）はステップＳ２００３で作成された色塗りアウトライン、（ｃ）はステップＳ２００３で作成された罫線の例をそれぞれ示す図である。
【図３３】図３２に示す（ａ）と（ｂ）を組み合わせて生成されたアウトラインによる表枠を示す図である。
【図３４】図３３に示すアウトライン表枠と図３２に示す（ｃ）のステップＳ２００３で生成された罫線とを用いて表枠を構成する例を示す図である。
【図３５】セル一つ一つを矩形として表現する罫線と通常の罫線を示す図である。
【図３６】ドキュメント・アナリシス・アウトプット・フォーマット（ＤＡＯＦ）のデータ構造を示す図である。
【図３７】電子文書作成処理の全体の概略を示すフローチャートである。
【図３８】文書構造ツリー生成処理の詳細を示すフローチャートである。
【図３９】ページの構成とその文書構造のツリーを示す図である。

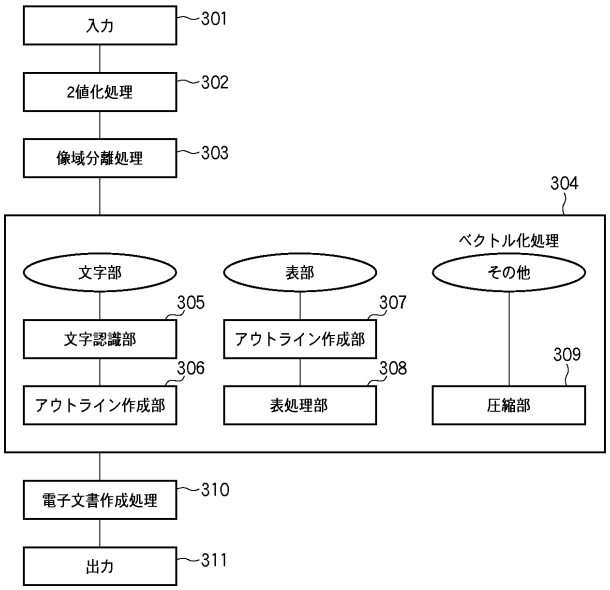
【図１】



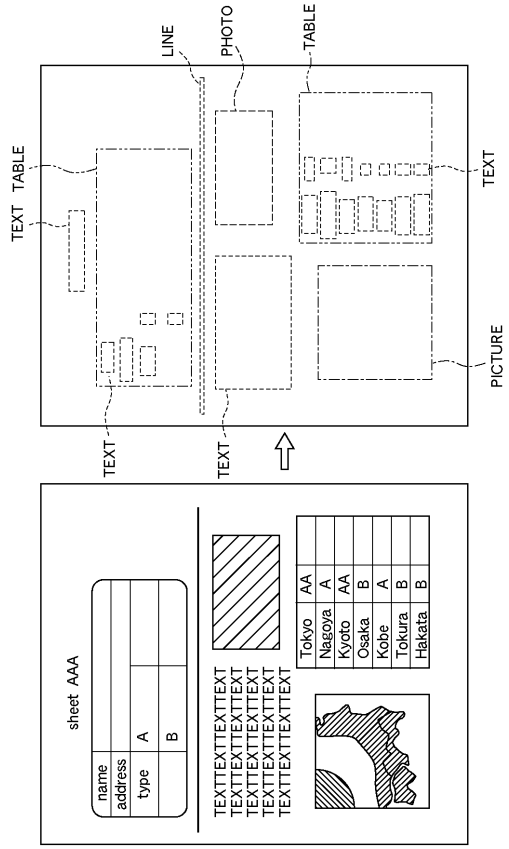
【図２】



【図３】



【図４】



【図５】

ブロック情報

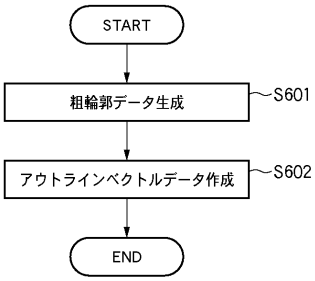
	属性	座標X	座標Y	幅W	高さH	OCR情報
ブロック1	1	X1	Y1	W1	H1	有
ブロック2	3	X2	Y2	W2	H2	有
ブロック3	2	X3	Y3	W3	H3	無
ブロック4	1	X4	Y4	W4	H4	有
ブロック5	3	X5	Y5	W5	H5	有
ブロック6	5	X6	Y6	W6	H6	無

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

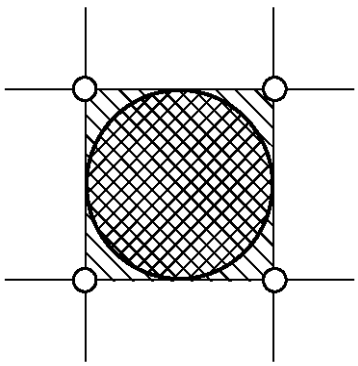
入力ファイル情報

ブロック総数	N (=6)

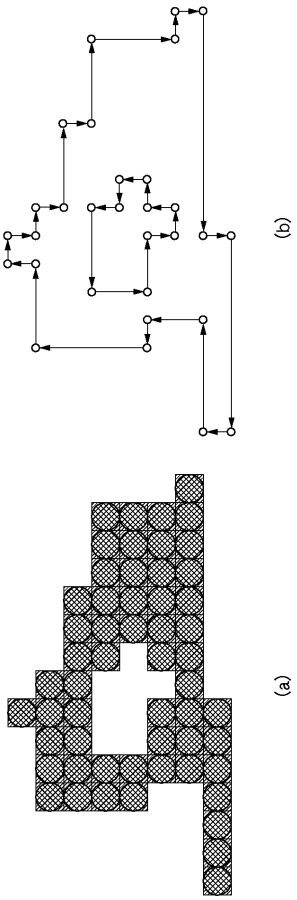
【図６】



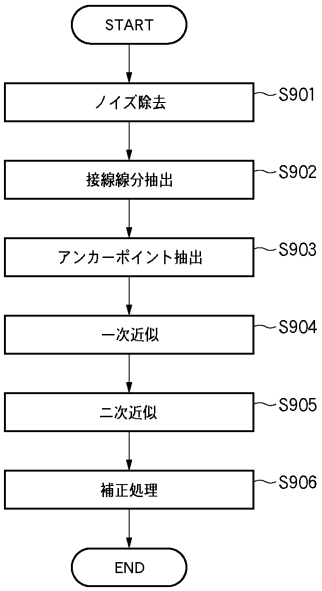
【図７】



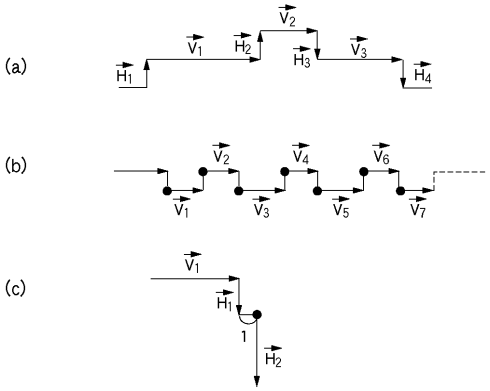
【図８】



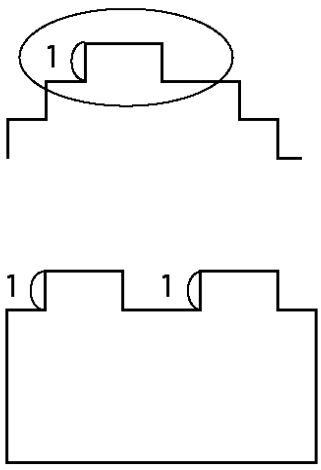
【図９】



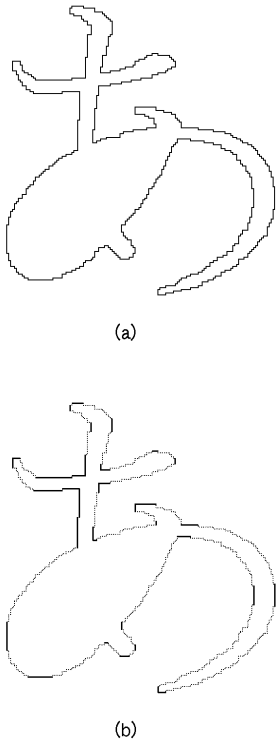
【図１０】



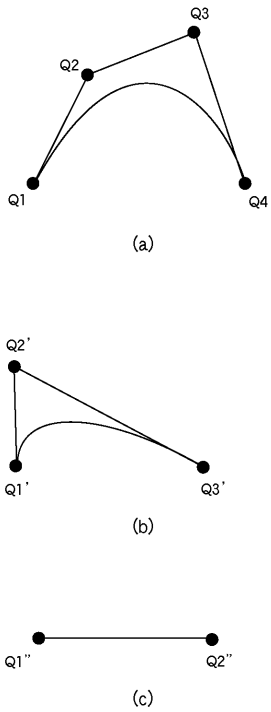
【図１１】



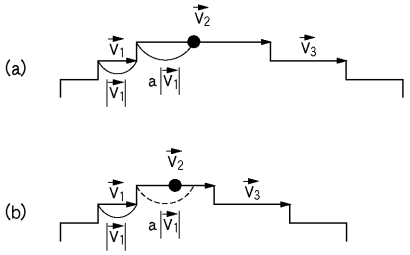
【図１２】



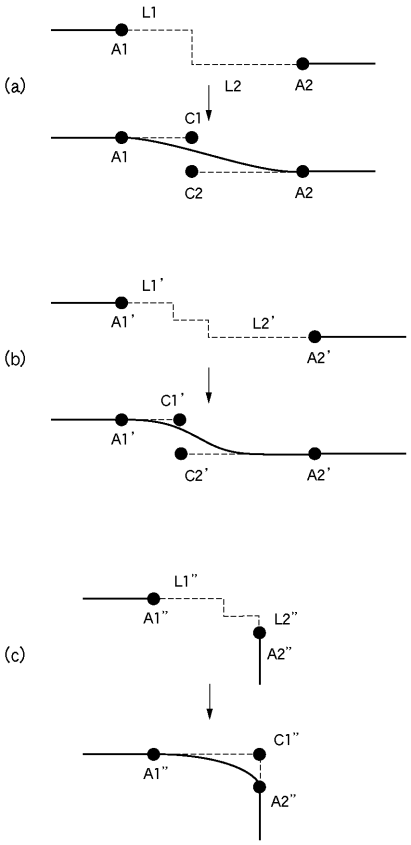
【図１３】



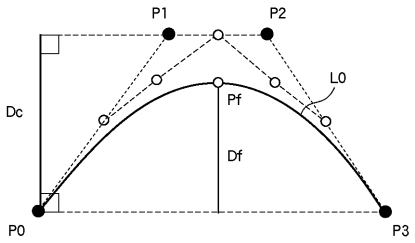
【図１４】



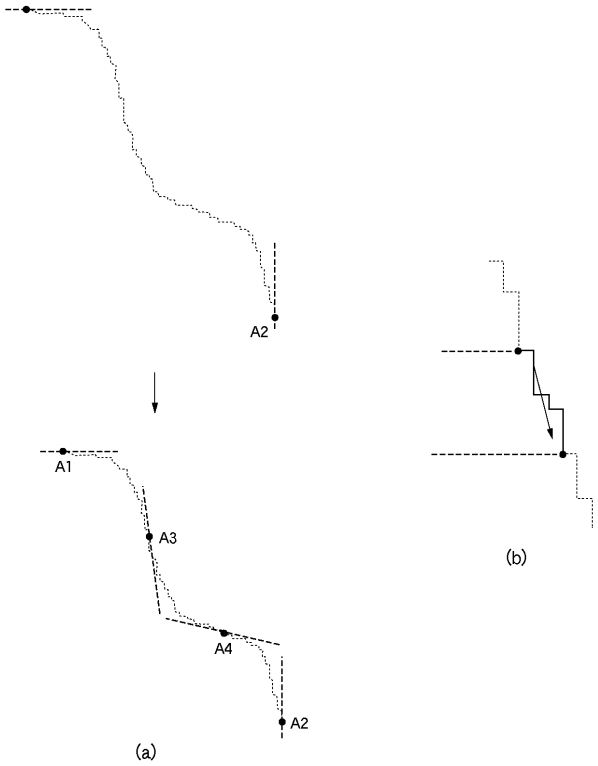
【図１５】



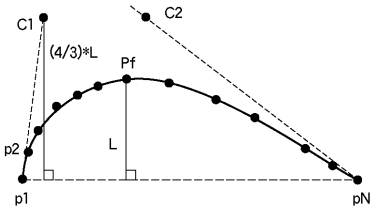
【図１６】



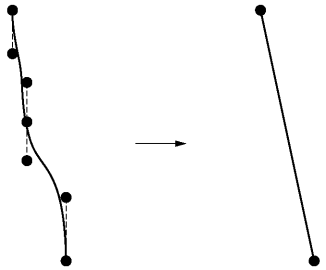
【図１７】



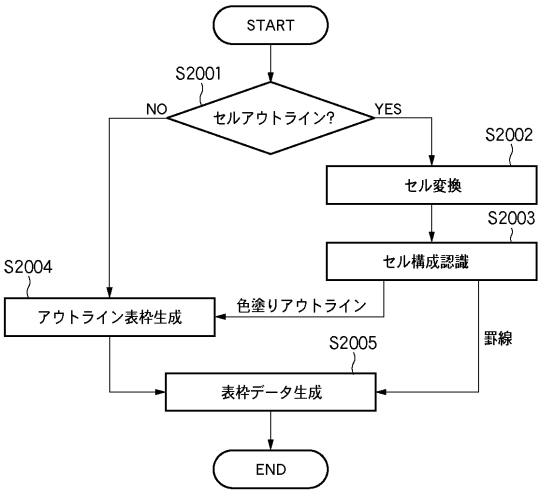
【図１８】



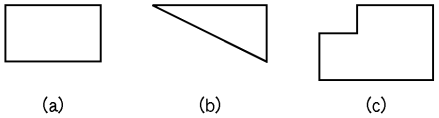
【図１９】



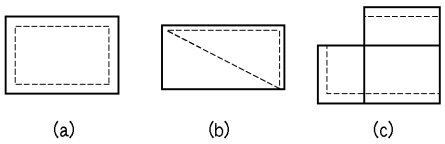
【図２０】



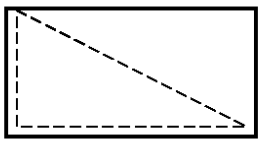
【図２１】



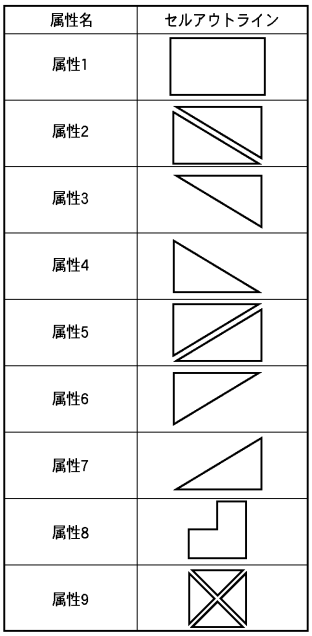
【図２２】



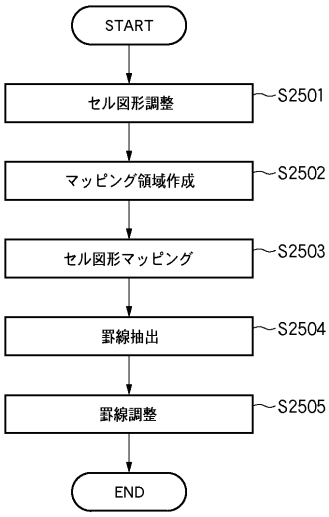
【図２３】



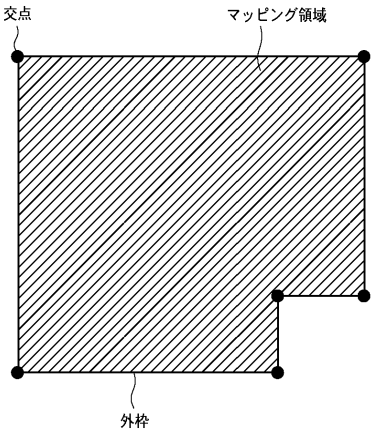
【図２４】



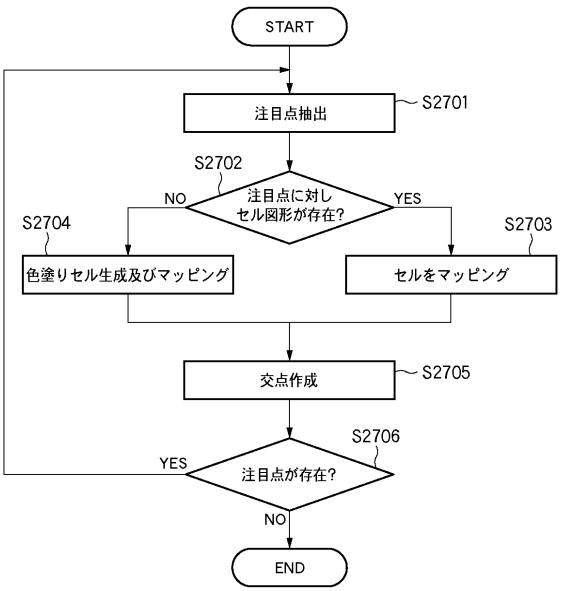
【図２５】



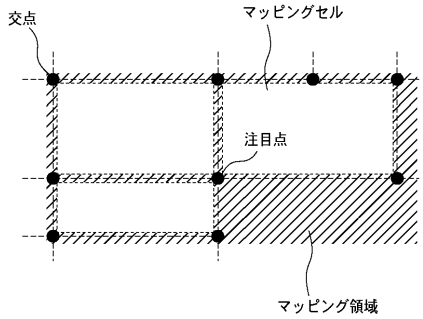
【図２６】



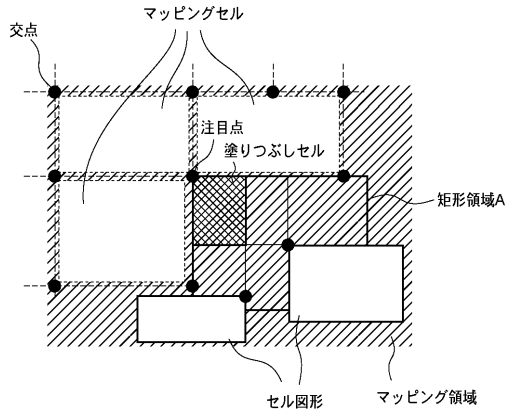
【図２７】



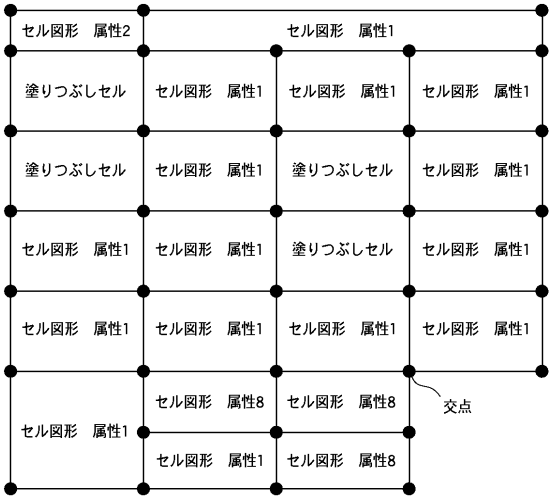
【図２８】



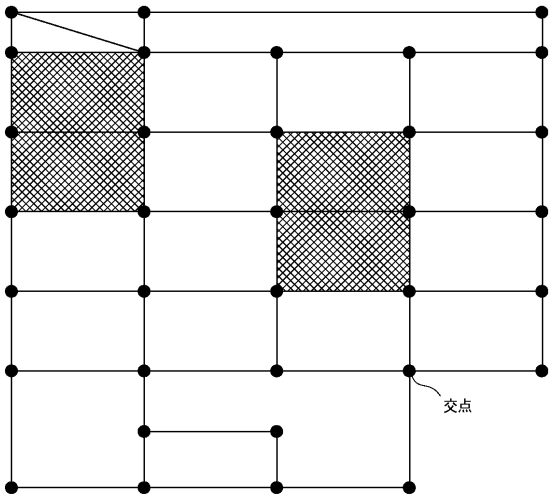
【図２９】



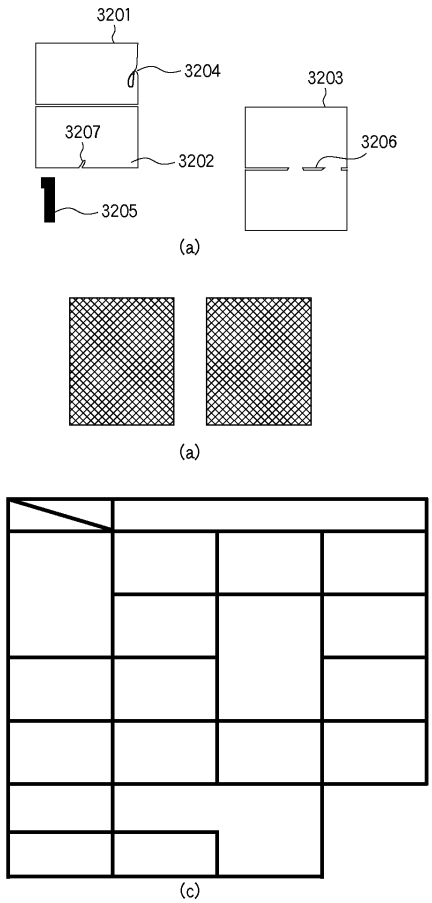
【図３０】



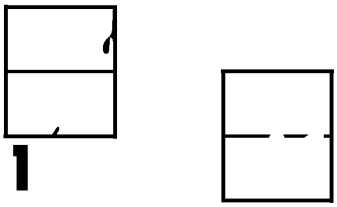
【図３１】



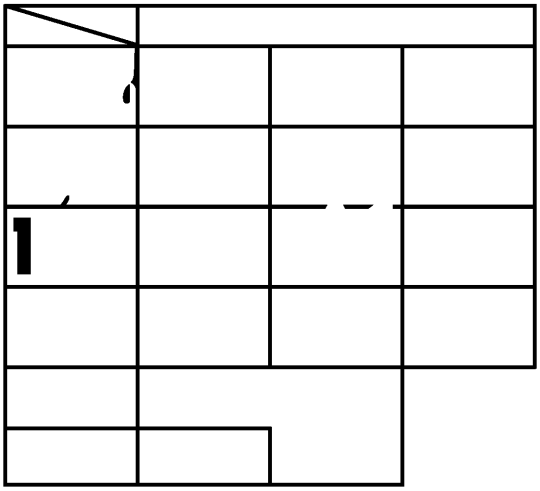
【図３２】



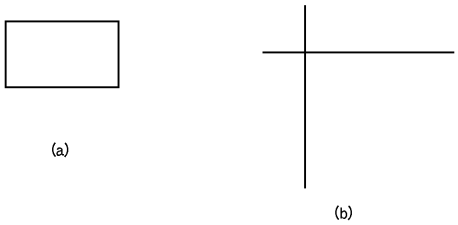
【図３３】



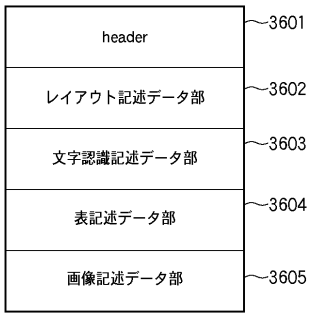
【図３４】



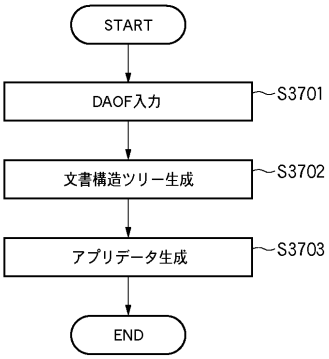
【図３５】



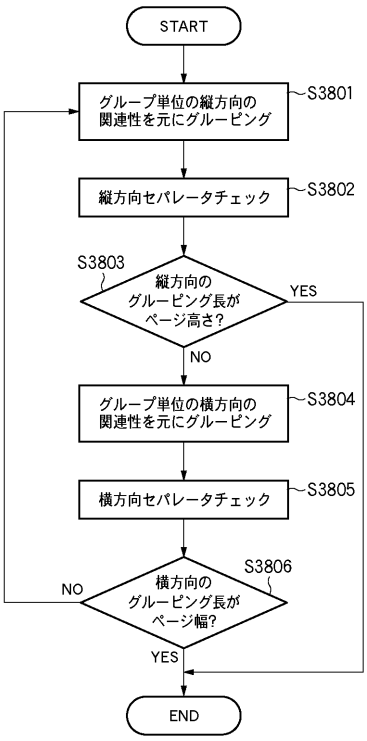
【図３６】



【図３７】

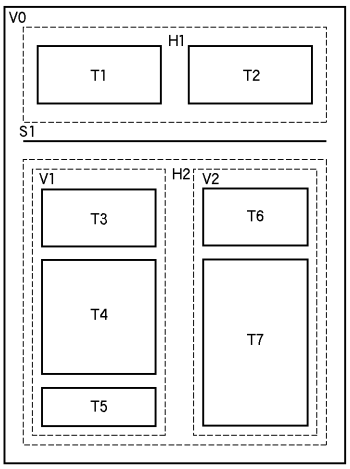


【図３８】

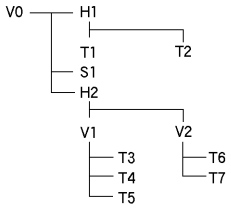


【図３９】

(a)


V0
H1
T1
T2
S1
V1
H2
V2
T3
T6
T4
T7
T5

(b)


V0
H1
T1
T2
S1
H2
V1
V2
T3
T4
T5
T6
T7