グラフィクスとCAD 35－3
（1988. 9. 29）

接触図形の分離を容易とする図面自動入力システム

大沢 裕　坂内 正夫
東京大学生産技術研究所

本稿では、筆者らが従来より開発を進めている図面自動入力システムAI-MUDAMSの芯線化方式の改良について述べる。基本方針は、まず図形の輪郭線追跡とその折れ線近似を行い、以後輪郭線データのみを対象として図形形状の抽出（芯線化）、図形の認識を行うというものである。従来の芯線化処理では、輪郭線に対しては何等手が加えられず、そのため認識・理解の段階で芯線と輪郭線の対応付けに多くの時間を必要とし、また処理の安定性にも問題を残していた。改良方式では、芯線化の際にすでに芯線作成の対象となった輪郭線ベクトルを切断・消去し、最終的に処理できなかった輪郭線、芯線及び芯線と輪郭線を対応付るglueベクトルの３者を作り出すものである。この改良により、芯線化後の図形整形や認識・理解が高速にかつ安定したものとなっている。

An Improvement of the Vectorization Method for Drawings
Facilitate the Separation of Overlapped Symbols

Yutaka OHSAWA and Masao SAKAUCHI

Institute of Industrial Science, University of Tokyo
22-1, Roppongi-7, Minato-ku, Tokyo, 106, Japan

This paper describes an improved version of the automatic drawing data capture system, named AI-MUDAMS. The basic concept of the system is as follows; object contours in the drawings pictures are first represented by contour vectors, and following procedures are performed using only these contour vector data. Several system have developed for many applications using this concept, including maps, electric diagrams and mechanical drawings. The old version, however, has problems on stability and time efficiency in the recognition process. The method proposed in this paper overcomes the shortcomings by combining core vectors with contour vectors, and make easy to detach symbols from line drawings.

1．はじめに

筆者らは従来より図形の輪郭線をベースにした図面自動入力システムＡＩ－ＭＵＤＡＭＳ[1]の開発を行っている。基本方針は、予め全ての図形の輪郭線を追跡し折れ線近似（ベクトル化）しておき、以後の処理は全てこの輪郭線データを参照しつつ行うというものである。このシステムで処理の中心となるのは、線図形の輪郭線から、その中心線を求める処理（芯線化処理）である。芯線化は線的な図形部に対してのみ行い、線の交点部分や他の図形との接触部では芯線は途切れさせたままで扱う。図面の認識・理解、会話的な図形修正に際しては、この「芯線が途切れている部分には図形を理解する上で重要な特徴点が存在する」という性質を利用することができ、現在までに機械パーツ図面[2]、地形図[3]などの自動認識・理解システムを実現している。

芯線が途切れている部分の周辺の図形（即ち輪郭線の形）を詳しく調べるための演算としては、途切れた芯線の端点を中心としてある大きさの探査枠を設定し、この枠内に何本の輪郭線と芯線が存在するか、輪郭線の形はどの様になっているかを、順に追跡して調べるという方法を採っている[4]。ＡＩ－ＭＵＤＡＭＳでは輪郭線ベクトルや処理結果として出力される芯線ベクトル等、全ての図形データを２次元的な図形データの管理に適したＢＤ木[5]と呼ぶデータ構造で管理している。そして、演算をこのデータ構造上で行うことにより図形同士の位置関係を主体とした各種演算の高速化が図られている。

しかし、従来方式の問題点は特徴的な部分の輪郭線や芯線の形状、位置関係などを詳細に調べる時の探査枠のサイスの設定法にあった。探査枠は図形中の認識しようとする部分を完全に包含し、かつタイトであることが望ましい。必要な部分を包含し尽くしていない場合には、探査枠を拡大して再調査が必要となることから、処理時間が増大し、また大きすぎる場合には複数の対象が枠内に入ってしまい、状況の判断が複雑になるという問題がある。

この問題を避けるため、芯線作成の都度、輪郭線上の対応する部分を消去していき、最終的には線の交点やシンボル部分の輪郭線ベクトルのみを残すという形に芯線化方式を改良した。この改良により、従来方式の利点を保存したまま、重畳シンボルの認識を容易かつ高速化することができる。

本稿では、まず２で芯線化の基本と従来方式の問題点を述べ、３で改良方式について述べる。４では新方式による処理結果について述べ考察する。

2．基本的な芯線化処理と問題点

2．１　芯線化処理

文献[1]で示した基本的な芯線化処理では、予め図形の輪郭線が追跡され、折れ線近似によりベクトルデータに変換される。そのベクトルデータをＢＤ木という階層的領域分割型の多次元データ管理構造で管理し、そのデータ構造上で次の手順で芯線化処理が実行される。

(1)未処理（まだ対応する芯線が発生されていない）輪郭線ベクトルの中から最長のものを１本取り出し、これをＶとする（図１(a)）。

(2)Ｖの進行方向に対して右側に存在し、かつ最も近い位置にあるベクトルＷを探す（この様に線図形の両側に対応するベクトルをペアベクトルと呼ぶ）（図１(b)）。

(3)ＶとＷの中心位置に芯線を発生させる（図１(b)の破線）。

(4)まずＶの進行方向に向かって、ペアベクトルＶとＷをたどりつつ芯線を発生する。端点や分岐点（これらはペアベクトル間の角度や連結関係により調べる）に達するまで(4)を繰り返す（図１

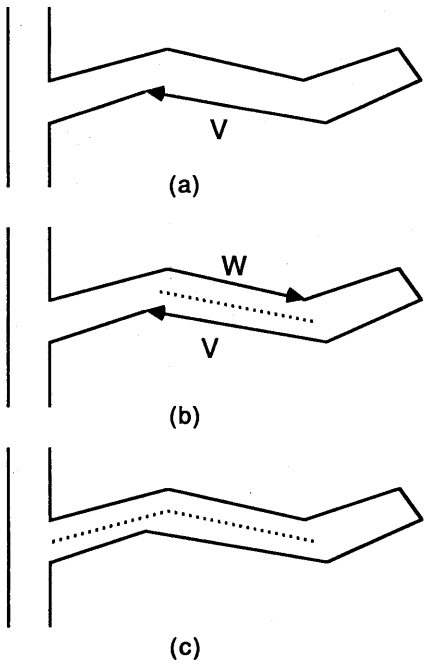


図1　芯線化の基本処理

(c)ではV，Wを左側に移動させる)。

(5)次にWの進行方向(図では右側)に向かって、(4)と同様にVとWをペアにしつつ芯線を発生する。

(6)(1)から(5)を、未処理の輪郭線の長さがL以下となるまで繰り返す。

(2)の処理でベクトルVの進行方向に対し、右側の探査を行っているのは、輪郭線追跡の際に図形の回りを時計方向に追跡し、輪郭線ベクトルに対してもこの方向性を与えている為である。これにより線的な部分の両側の輪郭線ベクトルは互いに逆の方向性を持ち、かつ互いに進行方向に対して右側に存在していることになる。

(6)で、芯線化処理を終了する際の未処理の輪郭線の長さLは図面の品質、内容に依存して定められるパラメータであり、線が滑らかで、形状が単純なものでは大きく定められ、逆に複雑に入り組んだ部分の芯線化を行おうとするときは小さく定められる。Lを小さく設定するほど処理時間は増大する。

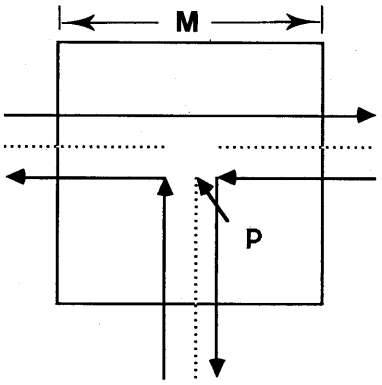


図2 芯線の結合処理

2．2　芯線の結合

前節で述べた処理により得られる芯線は全ての分岐点で途切れたものとなっている。これは、(4)(5)で芯線を発生していく際、端点及び分岐点に達したときにその部分における芯線化処理を中断し、他の部分から新たに種となる輪郭線ベクトルVを求め、その部分から芯線化処理を行う為である。従って、2．1で述べた芯線化処理後、途切れている芯線同士を結合する処理が必要となる。

この処理では、芯線の端点を中心とする正方形の探査マスクを用いて、端点近傍の輪郭線と、他の芯線の位置や形を調べ、それに応じた芯線同士の結合を行う。

図2の例を使って、芯線結合処理を述べると次のようになる。

(1)ある芯線の端点Pを中心とする、一辺Mの正方形領域(図中の太線枠)を設定し、その枠内の図形を対象として以下の処理を行う。

(2)端点Pに最も近い位置にある輪郭線ベクトルを探し、そのベクトルを種に追跡を行い、枠内に何本の輪郭線が入り、そして枠内から出ているかを調べその回数をCnとする(図2では3)。

(3)(2)で追跡された輪郭線により作られた芯線が枠内に何本存在するかを調べ、

その本数をＲｎとする（図２では３）。

(4)ＣｎとＲｎの本数により、状況を粗く分類し、各分類（例えば３交差点、４交差点等）毎に、より詳細に輪郭線の形状や芯線同士の傾き等により美的な、あるいは機能的な芯線同士の結合の仕方を決定し、結合する。

２．３　芯線化処理の問題点

次に、２．２で述べた芯線結合処理の問題点を考察する。

まず、(1)の正方形領域Ｍのサイズ設定が重要な問題となる。電子回路図にみられるように１枚の図面の中で図３(a)(b)の様な結合形が存在するとき、(a)に合わせて探査枠を設定した場合、(b)では交点部分を枠が包含できない場合が生じ、交差数（この場合には３交差点）を取り違える場合も生じる。一方、この様な状況を回避するため、探査枠を小さく設定した場合には、(c)に示すような状況で、極短い線分がＣｎやＲｎには表れてこず、交差点付近が状況を誤って判断されてしまう場合もある。

２．２で述べた処理は全て多次元データ構造、ＢＤ木上で行われるが、複雑に入り組んだ図形では枠内に多数の輪郭線ベクトルや芯線ベクトルが入ることになり、ＣｎやＲｎを求める計算は処理量の点で不利となる。

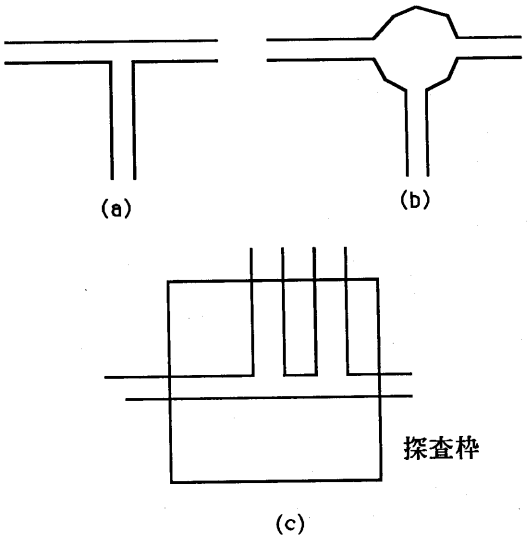


図３ 従来方式の問題点

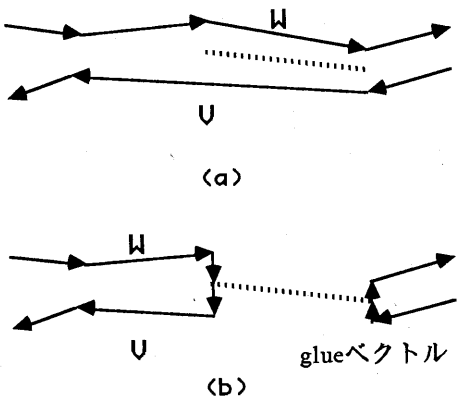


図４　輪郭線の切断と消去

３．芯線化処理の改良

３．１　基本方針

２．３で述べた以上の問題点に対処するため、輪郭線の扱いに対し次のような改良を行った。まず基本方針として、

①芯線作成時に用いられた輪郭線ベクトルを消去する。図４(a)に示すベクトルＶのように芯線作成時に一部分が対応する部分にのみ芯線が作られた場合、輪郭線ベクトルを分割し、発生させた芯線に対応する部分のみ消去する。

②芯線化を行って行き、分岐点や他の図形の重畳部分に達し、芯線化処理を中断する場合は、作成してきた芯線の端点と、まだ未処理の輪郭線の端点とを糊（glue）ベクトルで結合しておく。

このglueベクトルは図４(b)に示すように芯線の端点１つに付き２本発生される。１本は切断された輪郭線の端点から作成された芯線の端点に向かうものであり、また他の１本はその芯線の端点から他の輪郭線の切断点に向かうものである。glueベクトルにこのような方向性を持たせることにより全ての輪郭線がglueベクト

ルを含めて閉ループを形作らせられる。従って、芯線化後黒いベタ塗り領域が芯線化されずに残っても、その輪郭線はglueベクトルと共に閉ループの形で残ることになる。この性質は孤立シンボルがにじみにより線と接触している場合、そのシンボルを切り離し認識する際、などに有効である。

3．2　アルゴリズムの改良

改良方式の芯線化も基本的には2．で述べた芯線化と同様に行われる。相違点は、芯線を発生する際に処理対象となった輪郭線ベクトルを消去することと、芯線をそれ以上発生できなくなった分岐点部分で、芯線と未処理の輪郭線ベクトルをglueベクトルで連結する処理を行うことである。

(a)ベクトルデータの表現

輪郭線、芯線、glueの全てのベクトルデータは、始点・終点の座標と、線の種類を表すflag（輪郭線：1、芯線：2、glue：3）及びベクトル間の連結関係を表すポインタfrom、toの組で表現されている。また全ベクトルは階層的領域分割型多次元データ管理構造ＢＤ木により管理されている。

最初は全てのベクトルは、図形の輪郭線ベクトルであり、処理が進むに連れてそれが芯線やglueに置き変わっていく。ＢＤ木はこの様に動的なデータ管理に対し優れた性能を有している。またＢＤ木には輪郭線ベクトルの長さの長いものから順に1本づつ取り出す機能も具備されている。

(b)芯線化処理

まず、輪郭線ベクトルの中から最も長いものを1本取り出し、これをＶとする。次に、Ｖのペアベクトルｗが探される。ペアベクトルの探査に関しては文献[1]と同様にＶの進行方向に対し右側に存在する、位置が最も近いベクトルＷとＶの角度を求め、閾値θｐ以上であれば、これをベクトルＶのペアベクトルとする。また、もしベクトルＶのペアベクトルが求められなかった場合には、次に長い輪郭線ベクトルに対して同様な処理を試みる。

［Ａ：芯線ベクトルの始点の決定］

図５(a)に示すようにベクトルＶの直前、ベクトルＷの直後の輪郭線ベクトルをＶｐ、Ｗｎと表す。またベクトルＶの始点、ベクトルＷの終点座標を各々ｖｓ、ｗｅとする。作成される芯線の始点座標ｃｓをｖｓ，ｗｅの位置に応じて次のように決定する。

(i)ｖｓ，ｗｅ間の距離ｄｓが閾値Ｄｓより小さいとき（図５(a)）

ｖｓとｗｅの中点をｃｓとする。次にＶとＶｐ，ＷとＷｎのリンクを切り離し、Ｖｐの終点から点ｃｓへ向かうベクトルとｃｓからＷｎへ向かうベクトルをglueとして発生する（図中の破線がglueに対応する）。

(ii)ｖｓからＷに下ろした垂線の足がベ

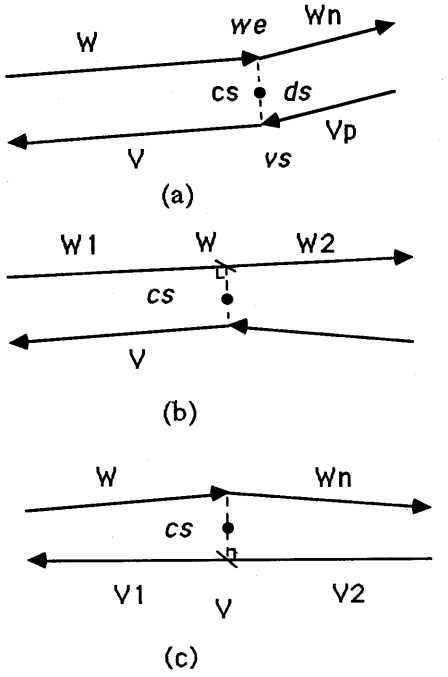


図５ 芯線の始点の決定

クトルW上にあるとき（図５(b)）

垂線の中点をｃｓとする。ベクトルWを垂線の足ｐの位置でW1、W2に分割する。ベクトルＶｐとＶのリンクを切り離し、Ｖｐの終点から点ｃｓに向かうベクトルとｃｓからベクトルW2の始点に向かうベクトルをglueとして発生する。W1を新たにWとする。

(iii)ｗｅからＶに下ろした垂線の足がベクトルＶ上にあるとき（図５(c)）

垂線の中点をｃｓとする。ベクトルＶを垂線の足ｐの位置でＶ1、Ｖ2に分割する。ベクトルWとＷｎのリンクを切り離し、Ｖ1の終点から点ｃｓに向かうベクトルと、ｃｓからベクトルＷｎの始点に向かうベクトルをglueとして発生する。Ｖ2を新たにＶとする。

［Ｂ：芯線ベクトルの終点の決定］

輪郭線ベクトルＶの終点ｖｅとWの始点ｗｓの位置関係により、芯線の始点ｃｓの決定と同様に終点ｃｅを決定する（図６参照）。この場合にもｖｅとｗｓの距離ｄが閾値Ｄｓより小さければｖｅ、ｗｓの中点をｃｅとし、それ以外の時ｖｅからWへ下ろした垂線及びｗｓからＶに下ろした垂線の足の位置により、終点座標ｃｅが決定される。

(i)ｖｅとｗｓの距離がＤｓ以下の時

ベクトルＶ、Ｗを各々消去する。Ｖｎを新たなＶ、Ｗｐを新たなW、ｃｅを新たなｃｓとして、芯線の発生を続ける。

(ii)ＶｅからWに下ろした垂線の足がW上にあるとき（図６）

ベクトルWを垂線の足ｐの位置でW1、W2に分割し、ＶとW2を消去する。Ｖｎを新たなＶ，Ｗ1を新たなW、ｃｅを新たなｃｓとして芯線の発生を続ける。

(iii)ＷｓからＶに下ろした垂線の足がＶ上にあるとき

ベクトルＶを垂線の足ｐの位置でＶ１，Ｖ２に分割し、ＷとＶ１を消去する。Ｗｐを新たなW，Ｖ２を新たなＶ，ｃｅを新たなｃｓとして芯線の発生を続ける。

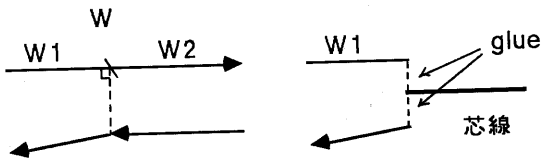


図６ 芯線ベクトルの終点の決定

芯線の終点の決定に際しては、「端点や分岐点などに達したか」という芯線発生の終了条件が常に調べられる。

(i)新しいベクトルＶ，Ｗが存在しないとき、または同一のベクトルとなったとき

芯線発生に使用したベクトルは、その都度消去していくため、線図形の端点ではＶｎやＷｐが存在しなくなる場合がある。

(ii)新しいベクトルＶ，Ｗの角度が閾値θtより小さいとき

分岐点やノイズ、シンボルなどの重畳箇所ではペアベクトル同士の成す角度が小さくなる。

(iii)新しいベクトルＶ，Ｗの少なくとも一方がglueとなったとき

図７に示すように、分岐点やループ状の図形を一周芯線化した際にglueベクトルに出会う場合がある。図７(a)の状態から芯線化が始まり、(b)の様に芯線化が行われてきたとき、次に芯線化を行い新しいＶ，Ｗを求めようとすると、Ｖ，Ｗは共にglueとなる。

Ｂの手続きを繰り返し、終了条件を満たしたとき、次に芯線作成の際に種となったペアベクトルＶ，Ｗから逆方向に同様に芯線化を行う。まず、いままで作成してきた芯線列の始点に連結されているglueを消去し、そのglueに接続されている２本のベクトルを新たなＶ，Ｗとして芯線化を行う。

以上で述べた芯線化方式の改良により、芯線間の結合や芯線端点近傍を中心とする図形の認識は単にglueベクトルで連結されている輪郭線の形状を調べるのみで

良く、特に図形同士が接触している場合の分離は、容易にかつ高速に実行することが出来る。例えば地形図で図８の様な部分の認識は、芯線の２つの端点a，bに連結している２つの輪郭線の形を調べ、一方が平坦、他方がコの字型という形状の特徴から、黒いベタ塗領域（家を表す）が接触していることが容易にわかる。

従来方式では、ＢＤ木上の近傍探査により芯線の端点aの近傍に存在する輪郭線ベクトルを探し、その輪郭線を追跡しながら近傍に存在する他の芯線の端点を探すという演算の繰り返しにより行っていたが、処理時間が長く、また安定した処理が行いにくいという問題があった。提案方式による改良により、高速でかつ安定した処理が実現されている。

4．処理結果と考察

本システムをデータゼネラル社のワークステーションDS7500上に実現し、性能評価を行った。1/500地形図の40cm四方を１ｍｍ当り１０本の分解能でディジタル化したものの処理時間の一例を示すと、ランレグス符号化されたディジタル図面データ上で輪郭線を追跡し、これを折れ線近似するのが７４秒、芯線化処理が８０秒であった。但し、ランレグス符号化された図面の輪郭線追跡にはPavlidisのアルゴリズム(6)を、また折れ線近似にはSklanskyらのアルゴリズム(7)を用いている。

そのほか多数の図面に対して、従来（文献[1]）の芯線化方式と新方式との比較を行った結果では、新方式の方が2、３割長い処理時間を要している。これは新方式では芯線化処理において輪郭線ベクトルを分割したり消去しており、それによりＢＤ木上での更新処理が従来方式より増大しているという理由による。しかし、その後の整形処理や認識処理では大幅な時間短縮と安定性の向上が達成されている。最も基本的な芯線の結合処理（文献[4]参照）に際しては、処理時間が１／３から１／５に短縮されている。これは芯線の端点近傍の結合状態を調査する際に、新方式ではＢＤ木上での検索処理の多くが不要となり、主としてリスト状のベクトル追跡処理のみで実行可能なためである。

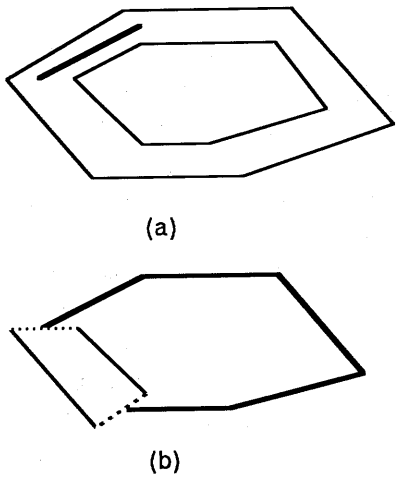
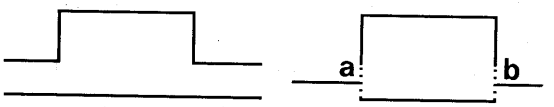


図7 ループ状図形の芯線化



(a) 芯線化前　　(b)芯線化後

図8 地図記号(黒マツ家屋)の芯線化

図９は1/25000地形図の地物版に対する処理の一例を示している。(a),(b)が芯線化処理結果を示しており、(c),(d)が各々交差点の処理、黒沫家屋の分離を行った結果を示している。従来方式では安定して切り出すことが難しかった(b),(d)の様な例も容易に実行できている。

現在、本芯線化処理をベースにした、1/25000地形図の認識・理解アルゴリズムの開発を行っている。

文献

(1)大沢、坂内：多次元データ構造を用いた図面処理－図形のベクトル化、信学論、J68-D,No,4、pp.845-852,1985

(2)魯、大沢、坂内：輪郭線を用いた機械設計図面の構造理解、第３５回情処全大、5K-3,1987

(3)滝嶋、大沢、坂内：自動入力図面のヒューマンフレンドリーな会話型修正システム、第３５回情処全大,7K-5,1987

(4)大沢、坂内：操作性を重視した自動入力図面の修正システム、第１７回画像工学コンファレンス、3-10,1986

(5)大沢、坂内：空間的な位置関係に依存した検索に適した線情報の管理方式、信学論、J69-D,No.5,pp.724-732,1986

(6)T.Pavlidis:Structural Pattern Recognition,Springer-Verlag,1977

(7)J.Sklansky,V.Conzalez:Fast Polygonal Approximation of Digitized Curves, Pattern Recognition, Vol. 12,pp.327-331,1980

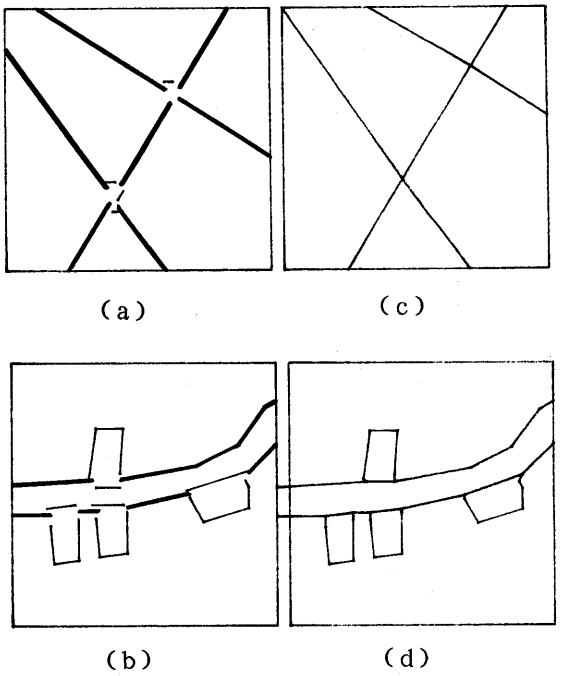


図9　1/25000地形図を対象とした処理例