

## 自動入力図面の自動整形と会話型修正システム

大沢 裕、滝嶋康弘、坂内正夫

東京大学生産技術研究所

図面自動入力システムAI-MUDAMSにおける、図面の自動修正と、会話型修正作業の効率化を考慮した図面修正システムについて述べる。AI-MUDAMSにおける図面入力処理の特徴は、線分の交点やノイズの重畳部分で必ず途切れた処理結果が出力されることである。従って、その出力の自動整形処理や会話型修正処理の基本は、途切れた線をいかに接合するかにある。自動整形処理では、この性質を使い芯線の端点の近傍の、他の芯線や輪郭線の形状を観察することにより、状況に合わせた接合を行なう。一般に図面の自動入力では文字の接触やノイズが重畳した部分等で図形に欠陥が生じる。大型図面の場合、このような場所を修正時にもれなく探し出すことは困難な作業となっている。本稿の後半で述べる会話型修正システムは、このような修正が必要な箇所をシステム側が人間に提示し、その部分に対してのみ適切な会話処理を行なうことにより出力結果の品質を容易に向上させることの可能なシステムである。

AUTOMATIC AND INTERACTIVE CORRECTION OF AUTO-DIGITIZED DRAWINGS AND MAPS

Yutaka OHSAWA, Yasuhiro TAKISHIMA and Masao SAKAUCHI  
Institute of Industrial Science, University of Tokyo  
7-22-1, Roppongi, Minato-ku, Tokyo, 106, Japan

An interactive graphic system for correction of the results from the automatic drawing and map input system named AI-MUDAMS is described. Since results which have been produced by automatic input system generally contains many errors, interactive correction is necessary to improve the quality of results. The proposed system is intended to provide good environment for this operation. It is difficult and time consuming work to find erroneous portions from automatic input data. This system has the function to point out these portions for ease of correction. The response of this system is also fast because of using the BD tree data structure for management of vector data.

## 1. はじめに

近年、設備管理図面、地図、各種設計図面のデータベース化の要求が高まるにつれ、図面の自動入力技術の重要性が増している。従来、図面の自動入力技術において解決すべき問題点の1つとして、処理時間の短縮があったが、これに関してはパイプライン処理や並列処理によるハード化や高速処理アルゴリズムの開発により大幅な短縮が計られ、実用的な時間内での処理が達成されつつある。しかし、多様な図面を対象にした自動入力システムの開発や、入力結果の質の向上、入力図面の完全さの検証等、まだまだ解決しなければならない問題が残されている。

図面の自動入力技術は、対象とする図面の種類により、次の2つの範ちゅうに分類できるものと思われる。1つはフローチャートやプロセス図、論理回路図の様に比較的少種類のシンボルとその連結関係が重要な図面群であり、他の1つは地図や機械・建築設計図面の様に特にシンボルという概念がなく、形状を忠実に入力することが要求される図面群である。要求される入力結果の品質は、当然これら2つのカテゴリー間で異なったものとなる。例えば、前者に対してはあるシンボルが他のシンボルに誤って判定されたり、判定不可能としてリジェクトされる確率が低い入力方式が望まれ、後者に対しては原図面に対する忠実度や美的な品質が望まれる訳である。

入力結果が正確である事は極めて重要である。特に前者のカテゴリーに属す図面においては、あるシンボルが他と誤って入力された場合、それを利用する際に全く役に立たない結果にもなる。また後者の図面に対しても、本来接触していない2本の線がインクのにじみや入力時のしきい値決定の不適切さにより結合して入力されてしまったものや、本来1本の線がかすれにより途切れて入力された結果は利用時に重大な欠陥となる場合も多い。

筆者らは以前より、汎用計算機上で実用的な処理時間を達成することを目的に、自動図面入力システムAI-MUDAMS<sup>(1)</sup>の開発を行なっている。先にあげた前者のカテゴリーに属す図面群を対象としたものとしてAI-MUDAMS Recognizer<sup>(2)</sup>を、また後者のカテゴリーの図面を対象としてAI-MUDAMS<sup>(1)</sup>を、それぞれ開発して来た。本稿では、このAI-MUDAMSの出力を対象に、後者の図面、即ち特にシンボルの概念のない図面を忠実に入力することを目的に開発した自動入力図面の自動整形システム、及び自動入力時に混入する欠陥をもれなく探し出し、人間に提示し修正をうながすことにより、自動入力結果の品質を高めることを目的に開発した会話型図面修正システムについて述べる。

## 2. AI-MUDAMSの概要

図面入力システムAI-MUDAMSは、汎用計算機上で高速かつ柔軟な処理を行なうことを目的として開発したシステムである。AI-MUDAMSの詳細については既に報告済み<sup>(1),(3)</sup>であるので、以下にその後の改良点と全体の概要のみを示す。

図面をデジタル化した画像データは一般に膨大なデータ量となる。例えば、A1図面(約60CM×80CM)を、比較的粗目の分解能(1mm当たり10本程度)でスキャンしたとしても6000×8000画素(48M画素)のデータ量となり、実記憶容量の小さな計算機でこの画像データを対象に処理を行なうことは大きな負担となる。そこでAI-MUDAMSでは、処理対象のデータ量を圧縮する為にまず画像データ中から図形部分の輪郭線を抽出し、それを折れ線近似したデータを作成する。以後の処理はこのデータを対象に行なわれる。この処理により、図1のA1図面は2万本程度(約100KB)のベクトル(0.15mmの許容誤差で折れ線近似)データになる。

従来、この輪郭線データを求める処理は、白・黒2値の2次元配列の画像データ上で行なっており、長い

処理時間を要していたが、現在ではランレグス化されたデータ上でこの処理を行なっている。即ち、黒ランの端点を前後のラインで対応づけながらたどることにより、直接輪郭線データを得る構造に改良している。この改良により、描かれている図形の少ない図面に対しては大幅な時間短縮を達成している。例えば、図1のA1図面に対しては5～6分程度の処理時間（DG社MV/4000）で輪郭線の折れ線データを得ることができる。

次にこの輪郭線データを線の部分と、孤立した文字、記号の部分に分類する処理を行なう。この処理は単純に連続する黒画素の塊の大きさを調べ、大きなものを線図形、小さなものを文字・記号に分類している為、線図形に接触している文字や記号は線図形の一部となり、かすれにより短く途切れた線図形の一部は文字・記号のカテゴリに分類されている。これらは、自動修正処理及び会話的修正処理により正しく再分類される。

次にこの線図形の部分を対象にして、輪郭線データからその中心線を求める処理を行なう（図2）。この処理結果で得られるデータは、画像レベルでの細線化を行ない、その処理結果を折れ線近似したものと同様のものであるが、通常の細線化処理と区別するためこれを芯線化と呼び、作られる線を芯線と呼んでいる。

図形中の1本の線的な部分に対し、輪郭線は線的なインク部分を挟んで2本存在する。この、互いにインク部分を挟んで存在する2本のベクトルをペアベクトルと呼ぶ。芯線化処理は、適当な1本の輪郭線ベクトルを選び出し、そのペアベクトルを求め、それらの中間に線を発生する処理である。ここで全体の処理時間を現実的なものとする為には、隣接するベクトルを高速に探査することが必要である。この為、AI-MUDAMSでは輪郭線や芯線のデータ管理構造としてBD木(4)を用いている。BD木とはベクトルの位置関係に依存した検索を高速に実行する為のデータ管理構造である。

この芯線化処理は、図面上の単純な線的な成分に対してのみ行なわれる結果、分岐点やノイズの重畳部分で必ず中断したものとなっている。そこで、次のステップとして、芯線の端点に注目し、その点の周囲状況に合わせて、必要であれば芯線同士を結合する処理が必要となる。これを、自動整形処理と呼んでおり、3.で述べる。

自動整形処理の出力はデジタル図面として完全なものとなっているのが理想であるが、現実にはノイズの重畳等により、システム作成時に予期されなかったパターンが発生する可能性もある。この様な部分に対

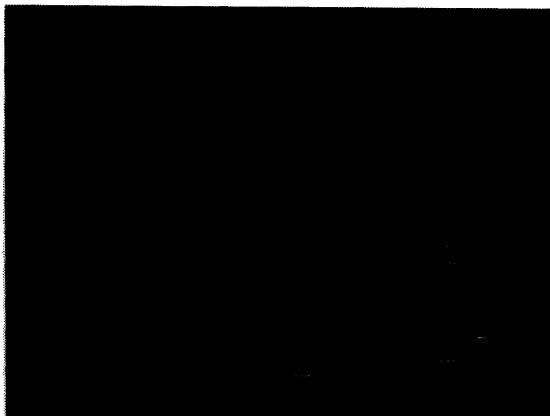


図1. 輪郭線のベクトル化データ

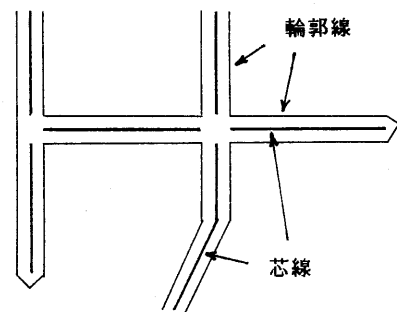


図2. AI-MUDAMSの芯線化処理

しては、会話的な修正システムを用いて人間が介在した修正作業が必要となるが、この際に問題となるのは、人間の負担が少なくかつ必要な修正箇所をもれなく探し出し、必要な修正を施すことである。4ではこの問題に対応する目的で開発中の会話型図面修正システムについて述べる。

### 3. 芯線結合の基本処理

先に述べた様にAI-MUDAMSの第2ステップでは線の交点やノイズの重畳した部分で途切れた芯線を作り出す。従って、完成した自動入力図面を得る為には、これらの点を判断し、必要な部分では結合する処理が必要となる。

これを、全ての芯線の端点において、その端点近傍の他の芯線や輪郭線を参照して周囲状況を判断し、それに合わせた処理を行なうことにより実行している。具体的には、図3に示す様に、芯線の端点を中心とする矩形領域を設定し、その矩形内の輪郭線や芯線が矩形枠と交差する回数を計数し、その回数によりこの部分の形状を判定し、それに合わせた整形を行なっている。図3の例では、枠には3本の芯線が交差しており、また輪郭線もこの矩形枠に3回出入りしていることから3交差点であることがわかる。この場合には、判定後3本の芯線のなす角度等により妥当な整形処理が行なわれる。

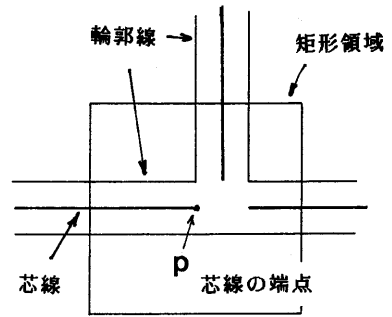


図3. 芯線端点に設定される矩形領域

#### 3.1 特徴量の抽出

自動整形の為のパラメータとしては次の2つのものを用いている。1つは輪郭線と枠との交差回数を表わすものであり、これを輪郭線交差回数と呼び、 $R_n$ で表わす。他の1つは、芯線と枠との交差回数を表わす特徴量であり、これを芯線交差回数と呼び $C_n$ で表わす。 $C_n$ 、 $R_n$ は基本的には、芯線の端点に対し、一辺Lの正方形領域を設定し、その枠内に何回輪郭線や芯線が交わっているかを調べることで得られる。またこの様な演算は芯線や輪郭線を多次元データ構造BD木で管理しておくことにより高速に実行できる。

少し複雑なものとして図4に示す状況を考えてみよう。この図に示されている枠内にはAとBで示す2つの独立した図形が描かれている。AとBはこの枠の外側で連結しているかも知れないが、この枠内、即ち点pの近傍では独立したものとなっている。今、グループA

に属す点pの近傍を考え、これを端点とするベクトルV1の連結を行なおうとする場合には、グループAに属す芯線及び、輪郭線に対してのみ枠との交差回数を数え、グループBに属すものは無視すべきである。この様な状況に対処する為、次の様な方法で $R_n$ と $C_n$ の計数を行なっている。

##### (1) 輪郭線交差回数

輪郭線の交差回数を求めるには、先ず注目している芯線の端点に最も近い位置に存在する輪郭線ベクトルを探し、その輪郭線を開始ベクトルとして、再度開始ベクトルにもどるまで、輪郭線ベクトルを順方向にたどることにより行なわれる。途中で、小矩形領域に達したら、時計回りに枠

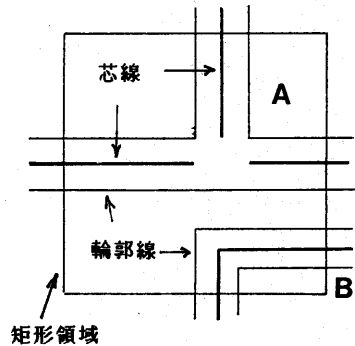


図4. 芯線と輪郭線のグループ化

上を探し、最初に見つかった枠の内側に入るベクトルをたどる。またこのとき、カウント回数を1加算する。最終的に再び開始ベクトルまで追跡した時点でのカウント回数が輪郭線交差回数である。具体的には、図4の図形に対しては点pに対し、最も近い位置にある輪郭線ベクトルとして、R\_Startが探され、以後矢印で示す順序でベクトルがたどられながら再びR\_Startに戻った時点で追跡が終了する。この処理を以下に示す手続きにより行なっている。

- (1) COUNT ← 0
- (2) 小矩形範囲の枠と交差する輪郭線ベクトルの集合Rを作る。
- (3) 注目している芯線の端点Pを中心とし、枠との交差角度によりRをソーティングする。
- (4) Pに最も近いベクトルを求め、これをR\_STARTとする。
- (5) r ← R\_START
- (6) rを枠の外に出るまで順方向にたどる。
- (7) 枠との交点QをキーにRを検索する。
- (8) Qより角度が大きく、かつ枠の内側に向かうベクトルを求め、新たにrとする。
- (9) COUNT ← COUNT+1
- (10) (6)-(9)のステップをrが再度R\_STARTに一致するまで繰り返す。
- (11) COUNTが輪郭線交差回数である

## (2) 芯線交差回数

芯線交差回数は次のようにして計数される。まず小矩形範囲内に存在する全ての芯線と小矩形との交点を求め、この芯線の中心Pとなす角度についてソーティングする。次に輪郭線交差回数の計算で作成した、輪郭線と小矩形範囲枠との交差位置の表を参照しつつ、芯線のベクトルが、現在注目している輪郭線の塊に含まれているかを判定する。この注目している輪郭線の塊に含まれる芯線ベクトルの本数が芯線交差回数である。

- (1) COUNT ← 1
- (2) 小矩形範囲の枠と交差する芯線ベクトルの集合Cを作る。
- (3) 注目している芯線の端点Pを中心とし、枠との交点の角度によりCをソーティングする。
- (4) Cの最初の要素から、それがRにおいて矩形から出るベクトルと入るベクトルの間にはいつているか調べる。
- (5) もし、矩形範囲から入るベクトルと出るベクトルの中間に存在すれば、COUNT ← COUNT+1
- (6) Cの要素全てについて終了した時点で、COUNTが芯線交差回数を示している。

## 3.2 芯線間の結合

この様にして得られたRn、Cnにより次の処理が行なわれる。

### (1) 端点の処理 (Cn=1、Rn=1)

線図形の端点の芯線データは図5(a)の様に、端点の輪郭から少し内側に入った所で終了している。芯線の端点は線の太さや線図形の端点の微妙な形状に依存して定まるが、理想的にはこれらに影響されない位置となることが望ましい。そこで図5(b)に示す様に、芯線を延長した線分に端点から最も近い輪郭線上の点からおろした垂線が交差する点まで芯線を延長する処理を行なう。

(2)断点の結合 ( $C_n=2$ 、 $R_n=2$ )

芯線は、図6(a)に示す様に、角部分やノイズの重畳箇所で途切れが生じることがある。このような場所では、図6(b)に示す様に、2本の芯線を延長した交点まで芯線を延長する。また2本の芯線が平行に近い場合には、単に2つの芯線の端点間を結合する処理を行なう。

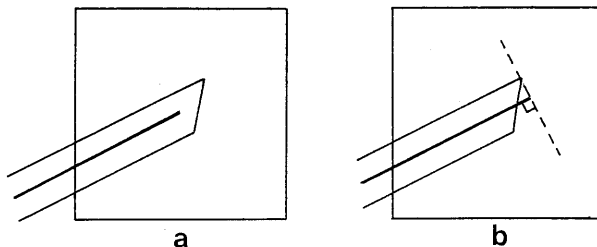


図5. 端点の処理

(3)芯線の長い欠落

( $C_n=1$ 、 $R_n=2$ )

ノイズの影響により、図7(a)に示す様な芯線の欠落も生じる。矩形範囲よりも長い区間で欠落している場合、輪郭線は連結しているにもかかわらず、芯線は矩形内で1本しか見つからないことになる。この際には図7(b)に示す様に輪郭線の形状に合わせて、矩形枠を移動しながら芯線を少しずつ延長する処理を行なう。

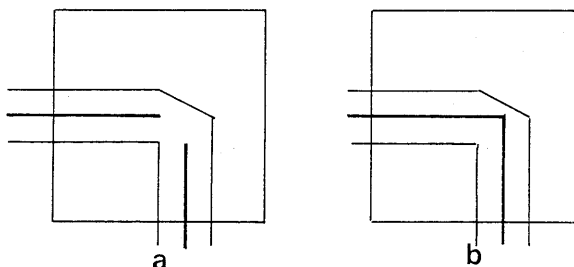


図6. 断点の結合

(4)3交差点 ( $C_n=3$ 、 $R_n=3$ )

図面上に表われる3交差点は、1本の直線に他の1本が交わっている形が多く見られる。そこで、3交差点の処理では、まず直線を構成している2本の芯線を探し出し、その2本を結合した直線と他の1本当の芯線の交点を求め、その交点に集まる様に3本の芯線を更新する処理を行なう。

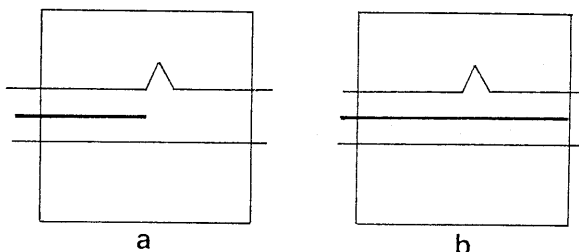


図7. 芯線の長い欠落

この他、 $R_n=C_n$ のときは $m$ 交差点となる。また $R_n \neq C_n$ のときは一般に、芯線の端点近傍の形状が複雑なものとなっていることを意味しており、自動整形の対象からははずされている。このような箇所は、次節で述べる会話型修正システムで処理される。

#### 4. 会話型図面修正システム

一般に、図面の自動入力で高い品質の出力結果を得る為には、品質の良い原図面を用意する必要がある。しかし、一般には自動入力用に図面をトレースし直すことはコストの高い作業となり、現在保有している図面を自動入力することが望まれる。しかし、品質の高くない図面を自動入力する場合、にじみやかすれ、しみ等の影響で出力結果に欠陥が生じ修正が必要となる。そこで必要となるのは、操作性の良い会話型の図面修正システムである。

この会話型の修正作業を行なう際の難しい問題点は、図面中から修正を必要とする場所がどこであるかを探し出すことである。一般にこのような作業は、原図面上に自動入力結果を色を変えてプロットし、不都合な

場所をそこから探し出すという形で行なわれるが、時間的にも心理的にも負担の大きな作業である。自動入力結果をディスプレイ上に写し出しながら修正作業を会話的に行なおうとする場合に、A1版程度の大型図面の場合、20インチ程度のグラフィックディスプレイに全体像を描いても非常に込み入ったものとなり、その中から欠陥部分を探し出すことはほとんど不可能となる(図1参照)。またズームアップしながら調べていく作業も時間のかかるものになる。

AI-MUDAMSでは、この作業を次の2つの工夫により容易なものとしている。1つは修正を必要とする箇所をシステム側が提示することであり、他の1つは原図面に近い形の輪郭線情報の必要な部分を高速に切り出して、処理結果に重畳表示することが可能なことである。図8に輪郭線を重畳表示した例を示している。この図では輪郭線を芯線の左上方に少しずらして表示している。これは完全に重ねて表示した場合、細い線の処理結果では輪郭線と芯線が重なり合い、どちらかが重書きにより消えてしまうことを防ぐ為であるが、位置を重ねて表示することも勿論可能である。

AI-MUDAMSの出力結果には、一般の図面入力システムと比べ大きな相違点がある。それは作り出される芯線データが、3.で述べた自動整形がなされていない状態では分岐点で必ず途切れているという点である。この途切れの多くは自動整形段階で結合されることになるが、ノイズが重畳したり、文字や記号と線が接触している場所ではまだ途切れたままになっている。従って、自動整形を行なった後、輪郭線が連続しているにもかかわらず、まだ途切れたままになっている箇所は会話修正を必要とする場所であることが分かる。また、図9に示すように輪郭線の形も線の端点となっている部分に対してはカスレが生じている可能性もある。AI-MUDAMSの会話型修正システムでは、自動修正段階で未処理のままとなっている、こ

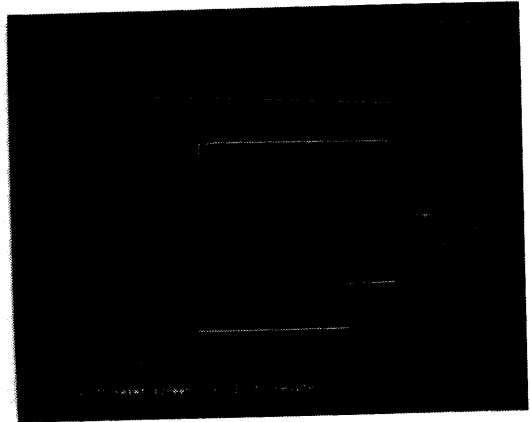


図8. 輪郭線の重畳表示

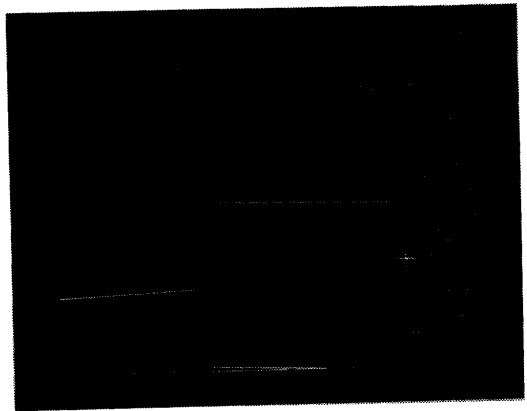


図9. 端点の確認

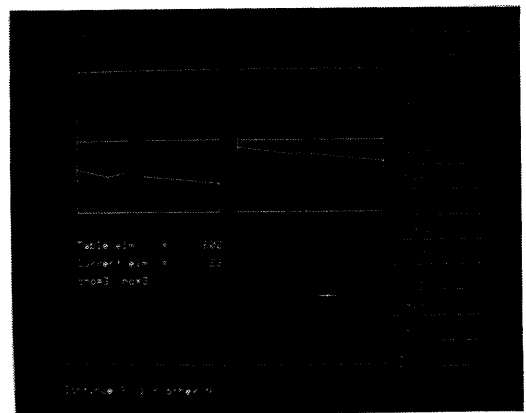


図10. システムによる修正箇所の表示

これらの点をシステムが探し出し、提示するモードを用意している。図10はシステムによるこの様な点の提示例であり、この場合には二本の横線に接続している縦線が極短かかった為、芯線が作られず自動整形段階で結合されなかった例である。この他に、大きな文字が描かれていて、処理の最初の段階で文字と判定されなかった箇所等も提示の対象となる。全ての芯線の端点が調べられた段階で、芯線データは品質の高いものとなっている。

本修正システムは、データ管理構造にBD木を採用しており、図形のピックやズーム等の会話に際しても速い応答時間で修正処理を行うことができる。

図11に本システムの作業画面を示す。画面左側にメニューが配置されており、上部4つが主メニュー、下部が現在の主メニューに関するサブメニューである。主メニューには上から「POLYGON」、「LINE」、「AUTO」、「UTIL」と書かれているが、「POLYGON」及び「LINE」が手作業による修正の為のモードであり「AUTO」は修正箇所を自動提示するモード、「UTIL」は各種ユーティリティーの為のモードである。

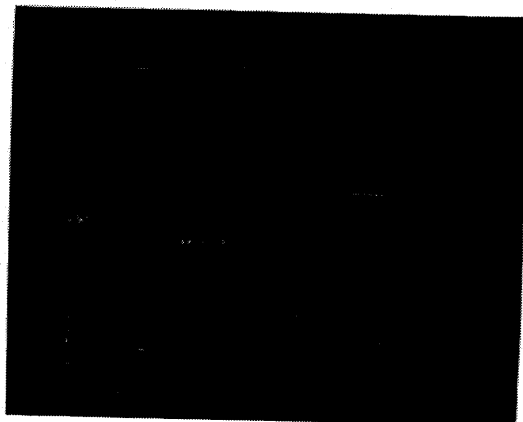


図11. 作業画面の一例

## 5. おわりに

本稿では、図面自動入力システムAI-MUDAMSにおける図面の自動修正と、会話型の図面修正システムについて述べた。図面自動入力システムで大量の図面を入力する際には、処理の高速化と共に可能な限り完全な形で入力することが重要となる。その際に不完全な部分を図面中からの的確に探し出し、その部分に対して修正を加える作業が特に重要となるわけである。本システムは、この様な作業の効率化をねらったものである。

現在のシステムはまだ完全なものとはなっておらず、各種の機能追加や会話作業の効率化が必要であるが現在これらの機能の実装を行なっている。また今後は同様の考えかたで、シンボル型の図面に対しても効率の良い会話型修正システムを開発する予定である。

## 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. 峯村、大沢、坂内：図面認識システムAI-MUDAMS Recognizerによる手書き論理回路図面の認識、昭61年信学総全大、1205, 1986
3. 大沢、坂内：多次元データ構造を用いた図面処理システムAI-MUDAMS、情処CV研究会市料、31-4, 1984
4. 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, 1986



**ICCV/87 • call for papers**

Non-profit  
Organization  
U.S. Postage  
Paid  
Silver Spring,  
Maryland  
Permit No. 1398

IEEE COMPUTER SOCIETY  
1730 Massachusetts Avenue, N.W.  
Washington, D.C. 20036-1903



OOOO  
ICCV/87

---

# Call for Papers

---

**FIRST INTERNATIONAL CONFERENCE  
ON COMPUTER VISION  
London, England  
June 8-11, 1987**



# Call for Papers

FIRST INTERNATIONAL CONFERENCE  
ON COMPUTER VISION  
London, England  
June 8-11, 1987

### COCHAIRS:

J. Michael Brady  
Dept. of Engineering Science  
Oxford University  
Parks Road  
Oxford OX1 3PJ, ENGLAND

Azriel Rosenfeld  
Center for Automation Research  
University of Maryland  
College Park, MD 20742

### PROGRAM COMMITTEE

Thomas O. Binford (USA)  
Jan-Olof Eklundh (Sweden)  
Olivier Faugeras (France)  
Takeo Kanade (USA)  
Jan Koenderink (The Netherlands)  
Hans-Hellmut Nagel (German Federal Republic)  
Tomaso Poggio (USA)  
Yoshiaki Shirai (Japan)  
Saburo Tsuji (Japan)  
Steven W. Zucker (Canada)

### THE CONFERENCE

ICCV is the first International Conference devoted solely to computer vision. It is sponsored by the IEEE Computer Society in cooperation with the International Association for Pattern Recognition (IAPR). It will be held in odd-numbered years, alternating with the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR).

### PAPER SUBMISSION

Four copies of complete drafts of submitted papers should be sent to *Azriel Rosenfeld* at the above address by *December 15, 1986*. Papers should contain major new research contributions. Authors will be notified of acceptance by early February, and final camera-ready papers, typed on special forms, will be required by early March.

### THE PROGRAM

The program will consist of invited review papers and high-quality contributed papers on all aspects of computer vision. All papers will be refereed by members of the Program Committee. To avoid conflicts, it is not planned to hold parallel sessions. A special feature of the program will be a set of invited talks on human perception and biological visual systems, as a continuation of the highly successful series of Workshops on Human and Machine Vision.

### Send to:

Azriel Rosenfeld  
Center for Automation Research  
University of Maryland  
College Park, MD 20742

### FOR FURTHER INFORMATION

Clip and mail to:

ICCV'87  
c/o IEEE Computer Society  
1730 Massachusetts Ave., N.W.,  
Washington, D.C. 20036-1903

Please send me further information on ICCV'87 as it becomes available.

Name \_\_\_\_\_

Affiliation \_\_\_\_\_

Address \_\_\_\_\_

City, State, Country, Zip \_\_\_\_\_



IEEE COMPUTER SOCIETY



THE INSTITUTE OF ELECTRICAL  
AND ELECTRONICS ENGINEERS, INC.

IEEE