

線画像認識方式 —文字と図形の混在する図面—

長田 茂美 井上 彰 吉田 真澄

( 株式会社 富士通研究所 )

1. まえがき

昨今のCAD/CAMの開発は著しく、多くの分野で実用化が進められているが、それに必要な図面情報の入力に人手に依存しているのが実状である。

線画像認識はこれを打開する最も有効な手段として期待されているが、今日の技術では、必ずしも現行のCAD/CAMに対処できない面が多く、まだまだ、数多くの解決すべき問題がある。

こうした中で筆者らは、図面入力の高速度化、および省力化を実現するために、特に、装置化につながる観点からの線画像認識方式の開発を進めており、既に、プリントパターン図に関しては、実用に耐え得る処理方式、ならびに、装置について発表した。(1)~(3) 本稿はそこで開発した技術を土台として確立した論理回路図認識方式に関するものである。

論理回路図には文字と図形(記号・線分)が混在しており、しかも、記号の種類が膨大であるという特質がある。このため、論理回路図を処理するためには、“文字と図形の分離”および“記号の識別”をどのように実現するかが最も大きな課題である。

このような問題に対し筆者らは、実用化、いわば、装置化に立脚した処理方式の検討を行い、その実現の可能性について確認した。以下、ここでは処理対象について触れた後、処理方式の基本的な考え方、特徴抽出、形状識別、特徴解釈の処理について詳述する。

2. 処理対象とその表現

図1は筆者らが処理対象として取り挙げた論理回路図の例、およびそれを表現するのに必要な情報を示す。

論理回路図は文字と図形から成っており、図形は論理記号、その組合せから成る複合記号、および各記号を結ぶ線分から構成される。図面処理にあたっては、文字(英数字36種類)はその存在する位置と文字コード、論理記号(60種類)および複合記号(30種類)は位置と予め定義されたコード名等を抽出する必要があり、また、線分は経路を表すベクトルで表現する必要がある。

なお、論理回路図はCADシステムへの入力を目的として描かれるもので、CADシステムに合致した座標系(格子)で表現することが要求される。

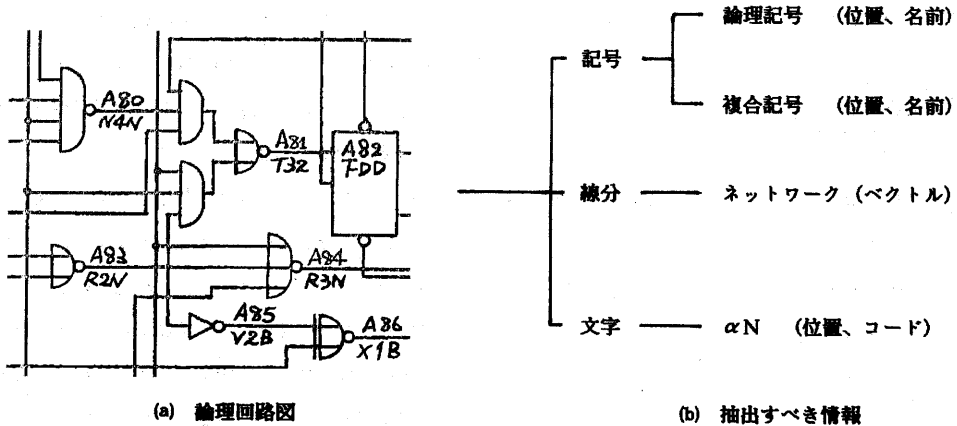


図1. 処理対象とその表現

### 3. 処理方式の基本構成

#### 3. 1 処理からみた図面の特徴

図1に示した論理回路図を線画像認識の観点から捉えてみると、次の特徴が指摘できる。

- (a)線分：4方向連結から成る直線である。
- (b)記号：種類が膨大である。また、寸法が規定されていないが、その大小の幅が大きい。
- (c)文字：書く位置が自由である。
- (d)構造：表現の異なる対象物が数多く存在する。

これらの中で、(a)は今日の処理技術からみて、さほど難しい問題ではないが、(b)~(d)に関しては、数多くの問題が含まれている。以下、装置化に立脚した方式を実現するにあたり、筆者らが特に留意した点について述べる。

##### <文字に関する問題>

図面の中に散在した文字の抽出は本質的に難しい。これまでに発表されている技術としては、図面の中から文字の属性を表すような特徴を捉え、それによって文字と図形の分離を実現しようとするもの<sup>(4)</sup>、あるいは、図面上の黒点を追跡し、その過程で文字を捉えようとするもの<sup>(5)</sup>等があるが、いずれも、処理時間、処理論理の点に問題がある。

##### <記号に関する問題>

◆種類…記号の種類が膨大なため、たとえ、図面の中から記号のみを抽出することができたとしても、それらを識

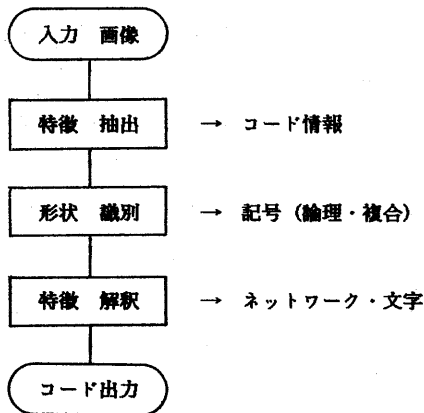


図2 処理の概要

別するための特徴量が膨大になる。

◆大きさ…大きさは特徴抽出の視野に影響を与えるが、論理回路図においてはその視野の設定範囲が膨大になる。

このように、記号の持つ特徴は、その特徴量、および特徴抽出の領域設定に問題を提起している。

##### <構造に関する問題>

表現形態の多様性は、各対象物を個別に抽出することの困難さ、さらには、対象物の関係記述の困難さを意味するものである。これまでの技術では追跡処理が主流であり<sup>(6)</sup>、これも処理時間、処理論理等の問題で装置化にはつながり難い。

#### 3. 2 処理の基本構成

図2は前節で述べた諸問題を解決するために開発した論理回路図面の処理フローである。ここでは各問題を以下の方策によって解決している。

##### <文字に関する問題>

図面処理の基本を信頼性の高い図形部分におき、「記号→線分→文字」という処理の流れを構築している。すなわち、図形部分の認識が終了すると、必然的に文字部分が抽出される構成になっている。

##### <記号に関する問題>

記号の構造を意識することなしに、図面全体を一定の論理条件下で簡単なコード情報に変換（特徴抽出）するとともに、形状の判定にあたっては、そのコード情報を用いた整合処理を採用している（形状識別）。これによって、特徴量は単純なコード情報の集合となり、また、大きさの幅は辞書内のモデル記述によって自由に表現できる。

##### <構造に関する問題>

図面全体をコード情報の集合に変換した後、それを基にして記号の識別、線分の記述（特徴解釈）、文字の抽出という三段階の処理によって各対象物を認識している。これは図面の中から直接、各対象物を捉えるのではなく、一連の処理の流れに沿って階層的に各対象物を捉えていくという立場である。なお、特徴抽出においては、各対象物間の関係を明示するような特徴も捉えており、ネットワークの

記述への配慮がなされている。

以下、特徴抽出、形状識別、特徴解釈の処理について詳述する。

#### 4. 特徴抽出

##### 4.1 処理概要

特徴抽出は入力画像を格子点を基準とした小さな矩形領域に分割し、各矩形領域内の形状に対して図形構造を捉え、それをコード情報として表現する処理である。特に、ここでは形状の直線性、連続性、および複雑さに着目した特徴を捉え、格子点単位に文字/図形らしさを表現するとともに、手書きによる変動に対処すべく、ズレの量を抽出し、図形としての形状の“確実さ”、“あいまいさ”も表現している。この処理のねらいは、入力画像の局所的な構造を圧縮表現したコード情報として記述することにより、後続の処理の効率化を計ることである。

具体的には、格子間距離  $n$  に基づき、 $2n \times 2n$  の領域を定義して、この領域内の図形構造を表現する“形状の4方向投影成分 ( $C_p$ )”、各方向に対する“形状の不連続性 ( $C_g$ )”、“手書き変動によるズレ度 ( $C_d$ )”、および“形状の複雑度 ( $C_c$ )”などの特徴を抽出するが、その処理は投影、正規化投影、矩形ウィンドウによる検証から成っている。

##### 4.2 抽出する特徴群

$C_p$  は矩形領域内の図形構造を簡単な4方向の形状で表現するもので、最終的に図面全体を記述するための基本要素となる。この特徴によって、矩形領域内の形状の概略は把握できるが、細部構造までは捉えられない。したがって、矩形領域内の形状が図形の構成要素であるか文字の一部であるかの識別は不可能である。しかし、 $C_p$  は投影という統計的な処理によって抽出されるので、手書きによる変動が吸取でき、格子軸に沿った図形の表現には非常に適している。

これに対して、 $C_g$ 、 $C_d$ 、 $C_c$  の特徴は矩形領域内の細部構造を捉えるもので、文字/図形を分離するための有

効な特徴となる。

$C_g$  は矩形領域内の形状が格子軸近傍において連続であるか否かを表現するもので、近接した文字と図形の境界の検出に有効である。

$C_d$  は矩形領域内の形状が直線成分を持つか否か、さらに、その直線成分が格子軸からどの程度ズレているかを表現するものであり、線分の極端な位置ズレの状態を検出するために利用される。

$C_c$  は矩形領域内の形状の複雑度を表現するもので、特に、文字部分のストロークの複雑さに敏感な特徴である。

このように、 $C_g$ 、 $C_d$ 、 $C_c$  の特徴は図面における文字と図形の本質的な相違点である形状の連続性、直線性、複雑さに着目しているため、文字/図形らしさを表現する情報として利用される。

以上の特徴群の抽出は、装置化への配慮から、水平、垂直の一次元走査および二次元ウィンドウ演算を基本とした単純な処理によって実現される。図3は特徴抽出の例を示す。 $C_p$  は投影処理、 $C_g$ 、 $C_d$  は正規化投影処理、さらに、 $C_c$  は矩形ウィンドウによる検証処理によって抽出される。<sup>(1)</sup> 図3(a)は投影処理例であり、(1)、(2)ともに左右、上方向の4方向投影成分が生成される。図3(b)は正規化投影処理例であり、(1)、(2)ともに下方向へのズレが検出され、(2)はさらに上方向への不連続性が抽出される。図3(c)は矩形ウィンドウの検証処理例であり、(2)の左右、上方向に複雑度が抽出される。

##### 4.3 格子点近傍の図形表現

特徴抽出の処理によって得られた  $C_p$ 、 $C_d$ 、 $C_c$  は上下左右の4方向に対応した4ビットで表現され、最終的には、文字/図形らしさを表現する情報  $F_f$  を付加した形で、各格子点毎に、格子点ラベルコード  $L(i, j)$  として統合される。

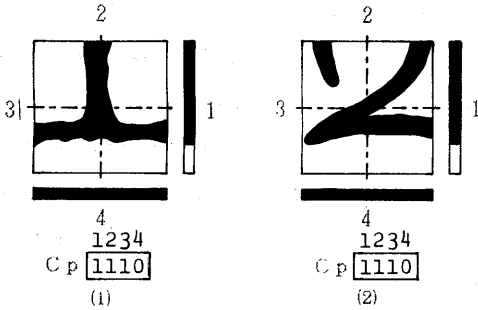
$$L(i, j) = [F_f, C_{pk}, C_{gk}, C_{dk}, C_{ck}] \quad (k=1, 4)$$

$$F_f = \begin{cases} 0 & : C_{gk}=0 \ \& \ C_{dk}=0 \ \& \ C_{ck}=0 \\ 1 & : \text{otherwise} \end{cases}$$

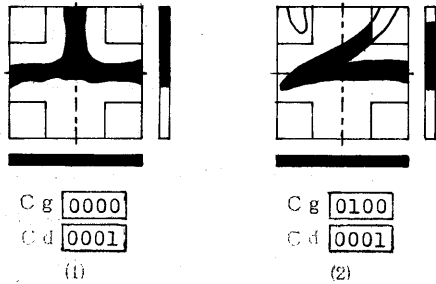
ここで、 $Ff=0$ は図形らしさを表現し、 $Ff=1$ は文字らしさを表現する。

こうして得られる格子点ラベルコードは格子点近傍の図形構造に、あいまい情報を加味した形で圧縮した図形表現であり、線切れ、ノイズに強い構造を持っている。

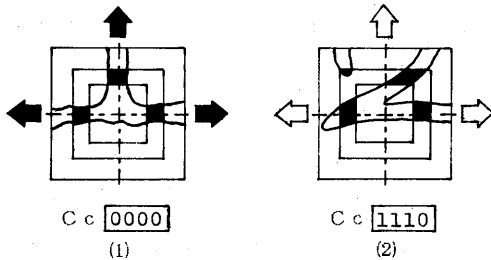
この段階で、入力画像全体は各格子点毎に圧縮表現されたコード情報の集合として表現される。



(a) 投影



(b) 正規化投影



(c) 矩形ウィンドウによる検証

図3 特徴抽出

## 5. 形状識別

### 5.1 処理概念

一般に形状識別では、あらかじめ識別対象となる記号の

内部構造を表現したモデルを設定しておき、そのモデルと特徴抽出によって得られた情報との整合処理を行うことを基本とした方式が用いられる。

しかし、論理回路図面内には記号として論理記号と複合記号が存在し、この両者は同レベルの関係にはない。つまり、複合記号は論理記号の集合体として位置付けされている。このような識別対象に対して、それぞれ同レベルのモデルを設定した場合、モデルの数が膨大となり整合に要する時間も増大する。

この問題を解決するために、本識別方式は二段階の階層構造をとっており、複合記号の識別は論理記号の識別結果を用いて行う形式になっている。これは論理記号と複合記号のそれぞれの性格に沿ったものであり、モデルの数も処理時間も減少する。

第一段階の処理は、論理記号の識別を行うもので、ここでは記号の内部構造を表現したモデルと特徴抽出によって得られた4方向投影成分との間で整合が行われる。この整合は類似度演算を基本としているため、手書きによる線切れやズレなどの変形による影響が回避でき、良好な識別結果を得ることができる。

第二段階の処理は記号相互間の相対位置関係を表現したモデルと第一段階で得られた結果との整合であり、これは複合記号を識別するためのものである。これにより、複合記号の識別に用いるモデルが非常に簡単になり、処理時間も短縮される。<sup>(7)</sup>

### 5.2 論理記号の識別

論理記号は手書きによる線切れやズレなどの変形が生じ易いため、ここでは、予め、識別対象物のモデルとしてあいまいな属性を付与した標準パターン(モデルI)を設定しておく。論理記号はそのモデルと抽出された4方向投影成分間の逐次的な類似度演算によって識別される。図4はルIの例である。

類似度 $S$ は、次式に従って計算する。

$$S = \sum \sum W_{xy}, \quad S \geq \theta \quad \rightarrow \text{論理記号}$$

ここで、 $W_{xy}$ は位置 $(x, y)$ で一一致した4方向投影成

分の数である。ただし、このとき、モデルの4方向投影成分があいまい情報の場合は一致、不一致にかかわらず加算するが、これにより、入力画像の線切れや変形に柔軟に対処できる。なお、Sが閾値θ以上の値をとるとき、対応するモデルが論理記号の種類を表現する。

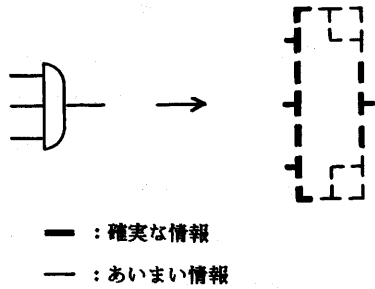
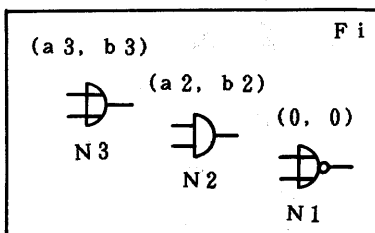


図4 モデルIの例

### 5.3 複合記号の識別

ここでは論理記号の集合体である複合記号の識別を行う。この識別にあたっては、論理記号の種類と相対的な位置関係からなるモデル（モデルII）を設定するが、このモデルの中には手書きによる位置ズレを考慮した位置（距離）関係を含ませている。図5はモデルIIの例である。ここで、N1、N2、N3は論理記号の名前を表し、(a2, b2)、(a3, b3)は、N1の記号の位置を原点(0, 0)としたときのN2、N3の相対位置である。図6は識別された複合記号の例であり、これは前節で述べた方法で識別された論理記号の集合とモデルIIとの整合処理によ



Nn : 論理記号の名前

Fi : 複合記号の名前

図5 モデルIIの例

て得られる。次式に示す条件がすべて満たされるとき論理記号N1、N2、N3のグループは複合記号Fiと識別される。

$$| (x_k - x_1) - a_i | \leq 2 \text{ (格子)}$$

$$| (y_k - y_1) - b_i | \leq 2 \text{ (格子)} \quad (k = 2, 3)$$

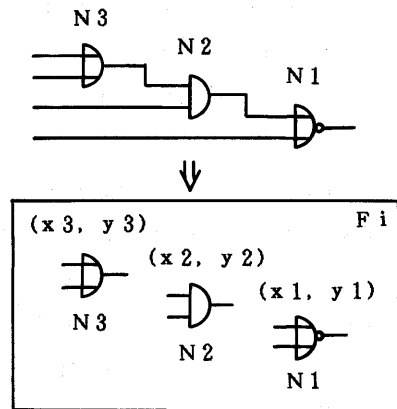


図6 複合記号の識別例

## 6. 特徴解釈

### 6.1 処理概要

特徴解釈は入力画像から抽出した特徴に関する知識を参照し、その知識から認識対象物の構造を仮定して、その仮定を実証するための他の特徴が画像中のしかるべき場所に存在するか否かを調べる。この過程を解釈しやすい部分から順番に試行錯誤的に実行することによって、図面全体の解析を進める。その結果、文字と図形が分離され、ベクトルから構成されるネットワークが抽出される。

解析にあたっては、特徴抽出によって得られたコード情報のうち、 $Ff = 0$ のコード情報を信頼性の高い確実な情報とみなし、この情報を基に、それに隣接する格子点に付与されたあいまいなコード情報 ( $Ff = 1$ のコード情報)の“あいまいさ”を順次確実な情報へと高めていくという立場をとっている。

特徴解釈は、線分のズレ補正、あいまい補正、文字/図形の分離の三つの処理から成っている。

## 6.2 ズレの補正

手書き図面の処理においては、線分の位置ズレを充分考慮する必要がある。本来単一の線分であるものが、この位置ズレに起因して、平行する二つの格子軸上のコード情報として二重に抽出される場合が生じる。

本処理はこのような線分の位置ズレの状態をコード情報から推定し、設計者が意図したと思われる格子軸上のみ、確実なコード情報を生成するものである。

図7はこの処理例であるが、最初に、相対する方向に“手書き変動によるズレ度 (Cd)”を有する一対の隣接格子を捉え、これを単一线分によるズレの表現と解釈する。次に、この隣接するコード情報を特徴 Cp に基づき逐次検索しながら、確実なコード情報が付与された格子の存在する軸を検出する。最後に、その軸を設計者が意図した格子軸とみなし、その軸上のコード情報を確実な情報へと高めるとともに、他方のコード情報を除去する。

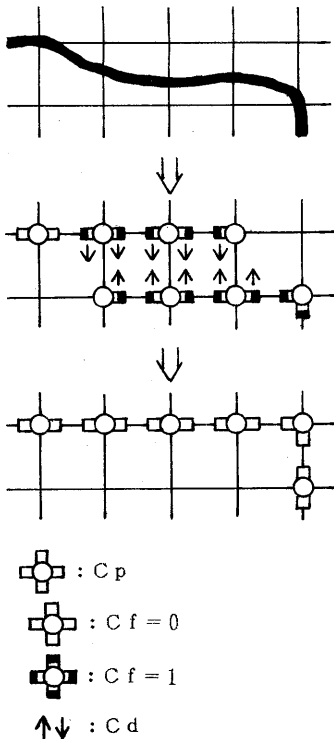


図7 ズレの補正

## 6.3 あいまい補正

本処理はコード情報の持つ“あいまいさ”をより大域的な視野から捉え、確実な情報へと高めていくもので、手書き図面特有の線切れ、雑音、あるいは線分に近接して挿かれた文字の影響による“あいまいさ”を除去する。

線分として不完全な形態を有する格子点には、“形状の不連続性 (Cg)”、“形状の複雑度 (Cc)”を持つあいまいなコード情報が付与される。当然、文字の存在する格子点にも同じくあいまいなコード情報が付与されるわけであるが、ここでは、このコード情報の持つ“あいまいさ”が、線分の局所の変動によるものか、それとも文字そのも

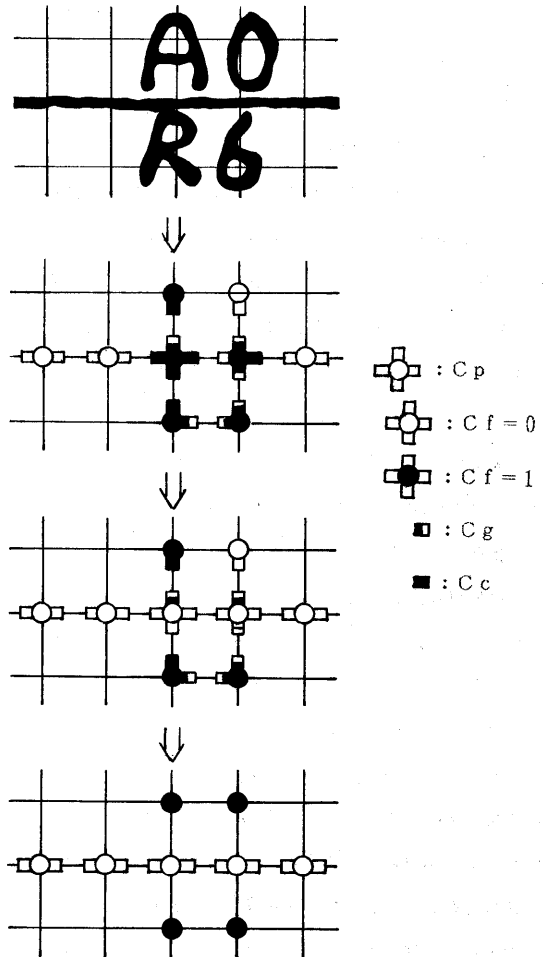


図8 あいまい補正

のに起因するのかの解釈を与え、その解釈に基づく補正を行う。

図8は本処理例であり、具体的には、最初に、あいまいなコード情報を検出し、その特徴 $C_p$ に沿った方向の格子点を逐次検索する。この検索の過程であいまいなコード情報を狭む形の確実なコード情報が出現すると、その“あいまいさ”は線分の局所的変動によるものと解釈し、確実なコード情報へと変換する。

本処理の結果、確実なコード情報は図形の構成要素として解釈され、最後まで“あいまいさ”を保持したままのコード情報は文字部分であると解釈される。

#### 6.4 文字/図形の分離

ここでは、格子点単位に付与されたコード情報を基に、文字/図形を分離し、線分の記述および文字領域の抽出を行う。

線分に関しては、形状識別の結果と確実なコード情報の $C_p$ を基にして、記号間を追跡し、ネットワークの構造をベクトルで表現する。

文字に関しては、あいまいなコード情報が付与された格子点近傍の画像データを矩形領域に切り出し、文字領域を抽出する。

#### 7. 実験結果

図9は図1の論理回路図を処理した結果である。各図はコード情報として得られた処理結果から復元したもので、図9(a)は特徴抽出によって得られた“形状の4方向投影成分( $C_p$ )”を示す。さらに、図9(b)は“形状の不連続性( $C_g$ )”、図9(c)は“手書き変動によるズレ度( $C_d$ )”、図9(d)は“形状の複雑度( $C_c$ )”を $C_p$ に重ねて表示したもので、各図において、太線は各々の特徴が抽出された位置および方向を示している。また、図9(e)、(f)は形状識別により抽出された論理記号・複合記号であり、図中の太線は複合記号を表している。図9(g)、(h)は特徴解釈の結果抽出された線分のネットワーク、および、文字領域を示す。

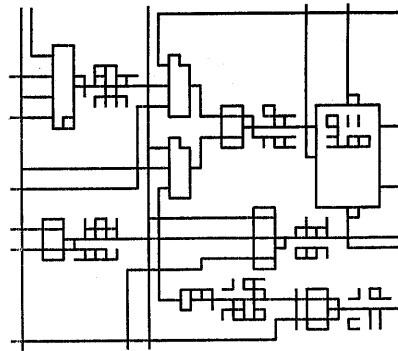


図9(a) 形状の4方向投影成分

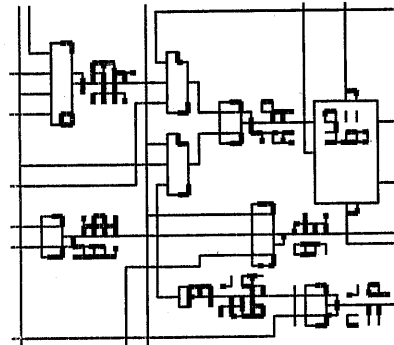


図9(b) 形状の不連続性

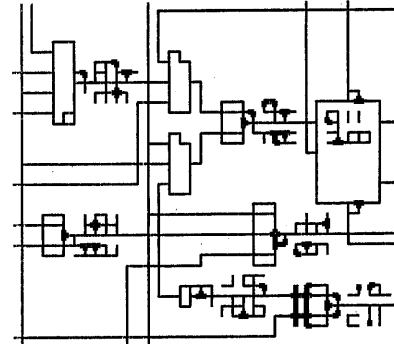


図9(c) 手書き変動によるズレ度

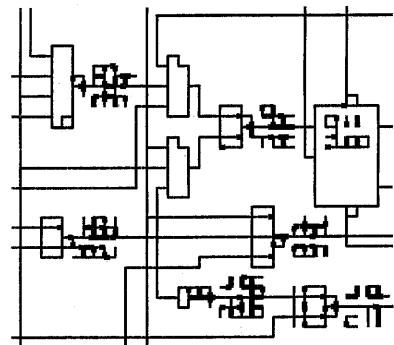


図9(d) 形状の複雑度

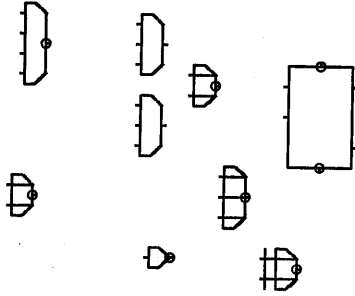


図9(e) 論理記号

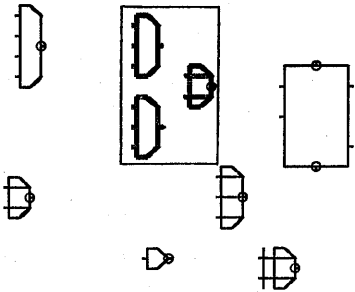


図9(f) 複合記号

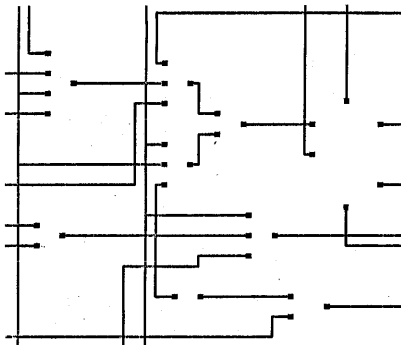


図9(g) ネットワーク

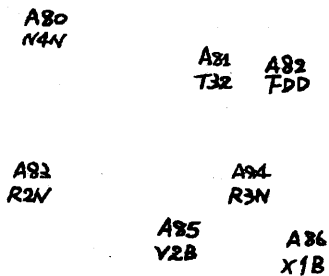


図9(h) 文字領域

## 8. むすび

本稿では線画像処理の例として論理回路図を取り挙げ、その中に混在する文字/図形を自動的に区分し識別する方式について述べた。

特に、ここでは装置化に直結した観点からの方式検討を行い、図面全体を一定の矩形領域単位で簡単なコード情報の集合に変換する“特徴抽出”、抽出されたコード情報とモデルとの整合により記号を識別する“形状識別”、およびコード情報間を逐次解釈しながら線分の記述・文字の抽出を行う“特徴解釈”の三段階の処理から成る方式を提案し、実際の図面に適用することによってその有用性を確認した。

なお、本処理方式は、現在、論理回路図面自動入力/処理装置として試作中である。

おわりに、日頃御指導いただく、宮川電子研究部門長代理、棚橋情報処理研究部長並びに関係各位に深謝する。

## 参考文献

- (1) 長田, 井上, 榊井, 松浦, 吉田: 線図形の読み取り方式, 電子通信学会研究会資料, PRL80-32 (1980)
- (2) 清水, 榊井, 長田, 松浦, 吉田: 設計図面の自動入力/処理装置, 電子通信学会研究会資料, PRL80-53 (1980)
- (3) 吉田, 榊井, 長田, 織田: 手書図面の自動入力/処理装置, 情報処理, Vol.22, No.4 (1981)
- (4) 井上, 吉田: 文字・図形分離方式の検討, 昭和56年度電子通信学会全国大会論文集, 1344 (1981)
- (5) 寺嶋, 首藤, 川井, 渡辺: 図面に書かれた文字の切り出し方式, 情報処理学会第23回全国学会予稿, 6C-4 (1981)
- (6) 佐藤, 棟上: 自由手書図面理解の基礎実験, 画像工学コンファレンス, 7-3 (1981)
- (7) 井上, 長田, 清水, 吉田: 類似度演算を用いた手書き図面内の形状識別, 情報処理学会第23回全国学会予稿, 4E-9 (1982)