

# 手書き図面清書システム FACORES

## Fair Copy Reproducing System for Hand-Sketched Diagrams

美濃導彦†・松井浩二†・坂井利之†・吉田裕一††

Michihiko MINO† Kouji MATSUI† Toshiyuki SAKAI† Yu-ichi YOSHIDA††

†京都大学工学部 ††㈱キャディックス

†Faculty of Engineering, Kyoto University ††CADIX Inc.

あらまし 我々は手書き図面を認識・理解し、清書するシステムFACORESを構築している。本報告では、FACORESのハードウェア構成及びソフトウェアシステムについて述べる。FACORESは(手書き)図面を入力すれば、その図面を理解し、入力図面に対応する清書図面を生成する。図面理解をした結果(図面の内部モデル)はCADシステムへの入力として利用できる。論理型図面を対象とした部分は既に完成しているが、現在は、図面の種類に関する制限を外して汎用的な図面処理システムとするために研究を進めている。

**Abstract** We are now constructing a system named FACORES (FAir COpy REproducing System) which understands a hand sketched diagram. In this report, we would like to discuss both the system hardware and the system software. FACORES accepts a roughly sketched diagram and produces the corresponding fair copy. The result of understanding the diagram can be used as an input to the CAD system. We have already constructed the part which can process a diagram representing a logical structure. And now, we are extending this system into a general purpose one.

### 1. はじめに

CAD/CAMの発展、普及に伴い、図面の計算機処理に対する要求が高まり、図面の対話入力手法や自動認識手法など、種々の研究、開発がなされている<sup>1)</sup>。図面は一般に種類が多く性質も大変異なる。図面一般に共通する性質は、それが線(筆点の軌跡)から構成されている線図形であるということ、及びほとんどの図面が文字列を含んでいること、などである。図面をそれに含まれる線の役割から考えると、大きく2つの種類に分けられる。その1つは、地図・製図などの「形状を表現した図面(形状記述型図面)」で、描かれている線

の形状が重要な意味をもつ図面である。他の1つは、電気・電子回路図、論理回路図、化学プラント図など、「論理構造を表現した図面(論理型図面)」で、図面を構成する要素の論理関係、接続関係を主として表現している。

我々は、図面の種類を限定せずに、(手書きの)図面を清書するシステムFACORES (FAir COpy REproducing System)を構築している。ここで、清書とは、図面を入力してそれに対応する認識結果の図面を得るまでの全ての処理を意味するものとする。

従って、清書処理は、一般的には、データ入力時に行う前処理としての入力時処理、画像上で点の系列として表された線をベクトル記述に変換するグラフ構造作成処理、グラフ構造を解釈してゆく認識・理解処理、及びその結果生成された図面の記述をもとに再び図面を生成するグラフィックス処理の4段階からなる。特に、対象が手書きの図面である場合は、図面シンボルを正確に一系列に並べたり、接続線を水平・垂直にするなど図面の記述に対する整列処理が必要となる。

上記各処理の中で、対象とする図面特有の知識を利用する処理は、認識・理解処理だけである。FACORESの中には現在のところ、論理型図面を対象とした認識・理解処理だけがインプリメントされている。本報告では、FACORESのシステム構成、及びその処理アルゴリズムについて述べる。

## 2 図面清書システムFACORESの概要

### 2.1 図面清書システムFACORESのハードウェア構成

京都大学工学部情報工学教室では、現在、媒体統合研究推進システムというプロジェクトが進められている。そのなかで図面処理に関係する部分を図1に示す。

ホストコンピュータNEC ACOS 850、MS 190、図面処理装置CADIX201がそれぞれ結合されている。NEC ACOS、MS間はOLA(On Line Adapter)接続で転送速度はDMAモードで約400KB/s、MS190、CADIX間は8ビットパラレル、20KB/s程度の転送速度でデータ転送が行える。研究環境は、アルゴリズムの開発、テストラン等の試行錯誤的実験は、ACOS、MSを中心とするホストコンピュータで行い、ある程度、アルゴリズムが固まればCADIX201及び対話処理装置(ワークステーション)だけでスタンドアロンシステムとして大量データを高速に処理する実験が行える。また、処理全体を信号処理部と解釈・理解処理部に分け、前者をCADIX201で、後者をACOS、MSで実行することもできる。

CADIX201は、図2に示すように大型図面を高速にベクトル化するための図形処理部を中心に各種スキャナ、プロッタ等の入出力装置、画像データ及びベクトルデータの表示、エディットを行うカラーグラフィックスディスプレイ、ホストコンピュータや対話装置(ワークステーション)との高速インタフェース等から構成されている。

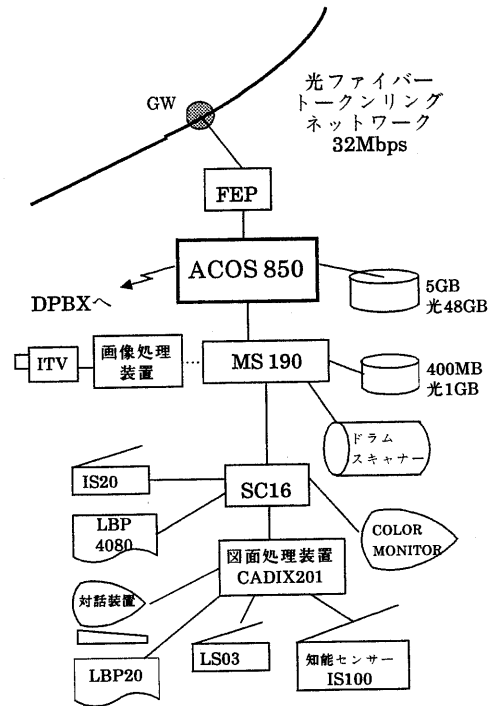


図1 システム構成図(全体)

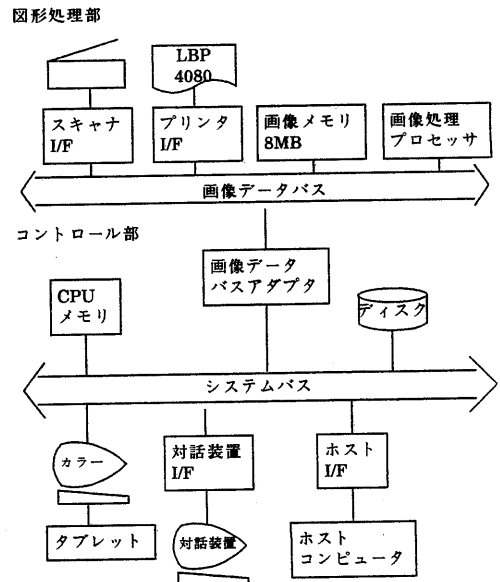


図2 CADIX201のハードウェア構成図

図形処理部はスキャナーからの図面データを画像メモリ部(8Mバイト)に格納し、各種画像処理専用のハードウェア(細線化ユニット、ベクトル変換ユニット、高速画像データアクセスユニット等)によってベクトル化を行う。コントロール部は、図形処理部及びシステム全体のコントロール、ファイル管理、オペレーション等の処理を行う。CADIX201は実験用のシステムであるため、入力装置としてはA3用リニアスキャナ及び研究室で試作した知能センサ、出力装置としてはA3用レーザビームプリンタを装備し、ホスト計算機に対する入出力コントローラとしても、ベクトル化前処理装置としても、またはスタンドアロンシステムとしても動作できるようになっている。

### 2.2 図面清書システムFACORESの処理の概要

図3に我々が作成した手書き図面の清書処理システムFACORESの処理の概要を示す。FACORESは図面の入力処理部、出力処理部とその間のデータ構造である“図面の記述”からなる。

図面の入力処理部は、入力時処理、グラフ構造作成処理、認識・理解処理からなる。我々が見ることのできる現実に存在する図面は、入力装置により入力されて計算機内でデジタル画像となる。従って、入力装置における前処理は、現実世界における図面と入力されたデジタル画像との同値性が保証できることが、最低限必要な条件である。図面は、人工情報源であり、線によって構成されているので、線に基づく記述法が基本である。ここでは、これをグラフ構造記述と呼ぶ。図面の入力処理部はグラフ構造記述を接点として図3に示すように大きく2つの部分に分けられる。1つは、信号処理の部分で画像として入力された図面から線を抽出し、そのグラフ構造記述を得る部分である。他方はグラフ構造を解釈し、対象図面固有の知識を用いて認識・理解処理を行う部分である。

対象が手書き図面の場合、線そのものがゆらぐこと、それぞれのシンボルの形状が正確に描かれないこと、及び、一列に並べるべきシンボルや特徴点などが正確な位置に配置されていないことなどの性質がある。線のゆらぎはグラフ構造作成処理で、シンボル形状の不安定性は認識・理解処理の中で対処すべき問題である。これに対して、位置揃えの処理は、独立な処理(整列処理)として清書システムに組み込む。

図面の出力処理は、整列処理が終わった図面記述を図面という人間の把握に適した形に変換するグラフィックス処理である。図面の記述が無矛盾であれば、この処理は単なる図面の生成処理であり問題は生じない。

図面の内部モデルは入力された図面に対して計算機が作成するものである。人間はこれに対して対話する。従って、内部モデルは図面の論理構造だけでなく、図面構成要素の配置に関する情報も含んでいなければならない。

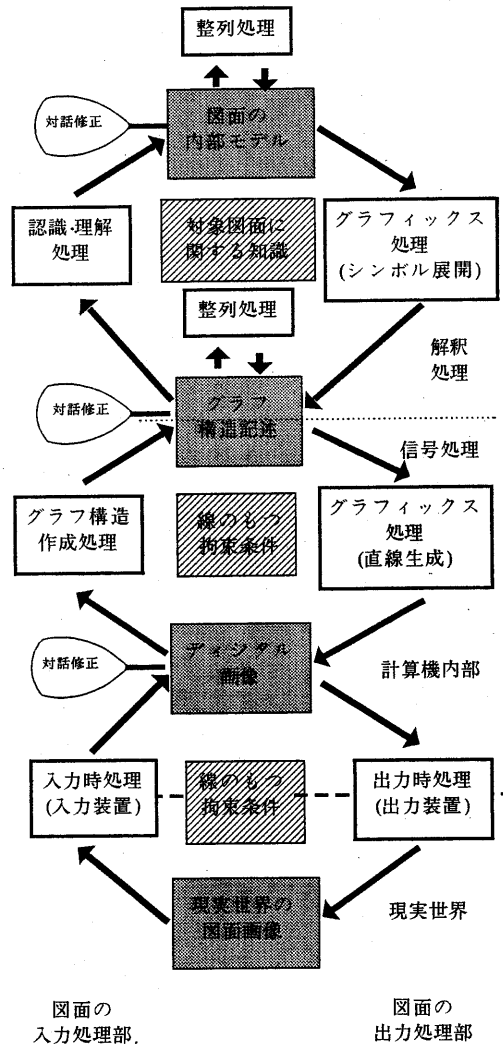


図3 図面の清書処理システムFACORESの概要

以下、FACORESにおける各処理について詳しく述べる。

### 3. FACORESの入力処理部

#### 3.1 入力時処理

現実世界における図面画像をデジタル化して入力する部分である。ここで問題となるのは、サンプリング時の解像度と閾値の2つのパラメータである。

解像度の問題は、基本的にはサンプリング定理により定まる。すなわち、対象画像が帯域制限されていて、そこに含まれる最高周波数が $W$ である時、 $1/2W$ の間隔でサンプリングすれば、もとの画像が復元できるというものである。Pavlidis<sup>2)</sup>が指摘しているように、この間隔でサンプリングした場合は二次元の位相的な性質が保存されない可能性があり、位相的に等価な画像を入力する場合は、 $1/2\sqrt{2}W$ の間隔が必要である。現実的には、この種の理論は二値化の閾値

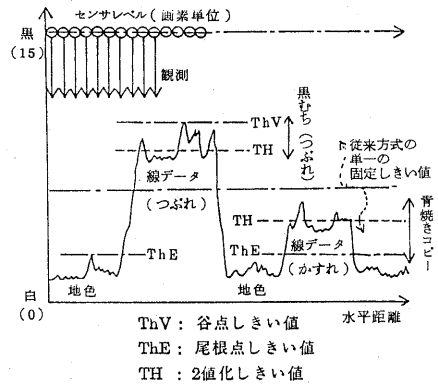
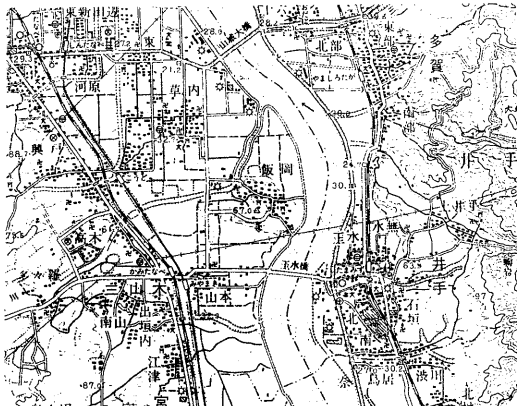
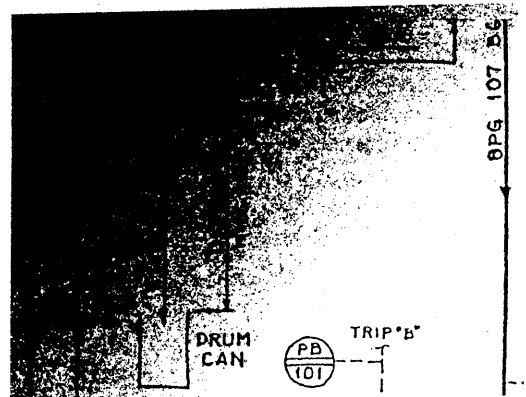


図4 尾根点・谷点の検出と二値化の原理

と密接に関連しているので、閾値の問題と同時に考えるべきである。



知能センサを用いない場合の入力図面



知能センサを用いた場合の入力図面

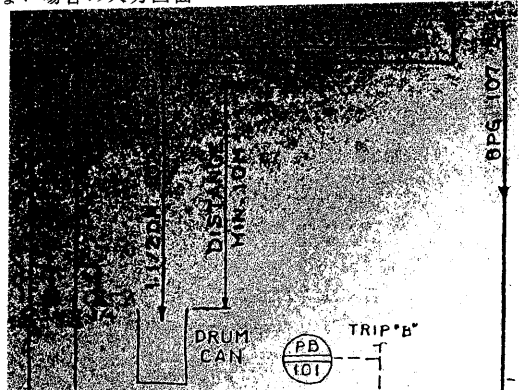


図5 知能センサによる入力例

実際に使われている図面は、品質の悪いものが多い。低品質図面を入力するための研究が始められつつある<sup>16)</sup>。

最も直接的な方法は、スキャナから図面を直接、濃淡データとして入力して処理するという手法である。しかし、図面は本質的には二値画像であること、サイズが大きく、かつ、細かい解像度が必要なことなどを考えると現在のところ、経済的に実現することは困難である。

現実的な手法としては、データの入力時に数行分程度の濃淡データを蓄積できるバッファメモリを持ち、局所処理で雑音除去などの簡単な処理を行うものが考えられる。我々は、この考え方に基づき知能センサ装置を試作した<sup>3)</sup>。この装置は、スキャナから濃淡データ(4ビット)を受け取り、2走査線プラス2画素分の濃淡データバッファを用いて局所的に“線”らしいところを3×3の窓内で尾根点・谷点として捉える。図4に示すように、尾根点はその絶対濃度に拘らず黒(線)とし、谷点は白(地)とする。尾根点・谷点が予め設定された複数(5つ)の閾値により黒(白)にならないところを尾根点(谷点)エラーとして検出し、エラーの総和が最も少ない閾値を動的に選択し、実時間で二値化するものである。図5に知能センサを利用した場合と利用しない場合の入力結果を示す。図面処理を行うという目的のもとで本当に効果があるかを現在検討中である。

知能センサに付与すべき他の機能として、写真領域の判別<sup>4)</sup>、低品質画像からの線の抽出<sup>5)</sup>、特定色の領域のみの抽出<sup>6)</sup>など、種々のものをソフトウェア的にシミュレーションを行っている。また、対象画像の画質を判定して、適応的な処理を行うこと<sup>7)</sup>や、文字だけを抽出する機能<sup>8)</sup>など、もっと高度な機能を付加して情報フィルタの作成を目指している。

### 3.2 グラフ構造作成処理

図面は、人間の意図を筆跡の軌跡として紙の上に表現したものであり、線により構成されている。図面を処理する時はこのことを十分利用しなければならない。入力時処理を経て入力された図面は、計算機内部では二値のデジタル画像として表現されている。図面処理の次のステップは、白黒画素として点の単位で独立に表現されている図面画像(画素レベルの記述)を、図面を構成する線を単位とする記述(ベクトルレベルの記述)に変換することであり、本稿ではこれをグラフ構造作成処理とよぶ。この過程は、信号処理で

あり、線のもつ拘束条件を利用して、局所的に線を把握することが重要である。

図面には、一般的には、線以外に黒塗り面も存在する。また、線そのものにも太さや種類(実線・破線・一点鎖線など)があり、その形状(直線・円弧・曲線など)も異なる。さらに、印刷されているか、手書きであるかなど、線の描き方にも大きな違いがある。

線の抽出処理は、大別すると2通りの方法がある。1つは、まず、線を細線化して幅が1画素の線とした後に、交点・屈折点などの特徴点を抽出し、ベクトル化する方法、他の方法は、線を細線化せずに直接特徴点を抽出する方法である。FACORESでは、グラフ構造(特徴点グラフ)による記述形式を明確に定義し、ここでの処理をコンポーネント化している。言い換えれば、種々の方式のグラフ構造作成処理を任意に組み込める枠組みを作成している。この枠組みの中で、我々は両方からアプローチしている。

#### (1) 線を細線化する手法

図面に含まれている線を細線化すれば、その交点・端点・屈折点などの線の特徴点を抽出する処理は簡単で正確なものとなる。図面処理装置CADIX 201では、この手法で高速に線をベクトル化するための種々のハードウェア・ソフトウェアが準備されている。

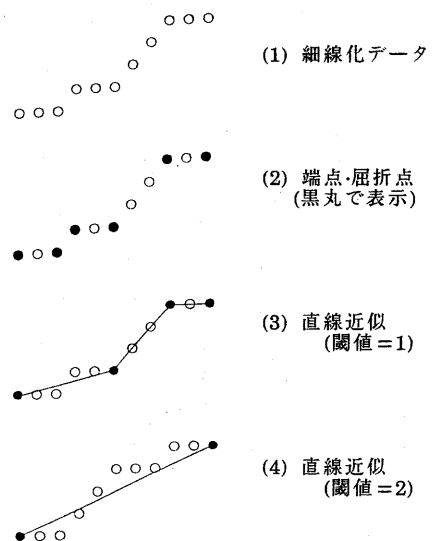
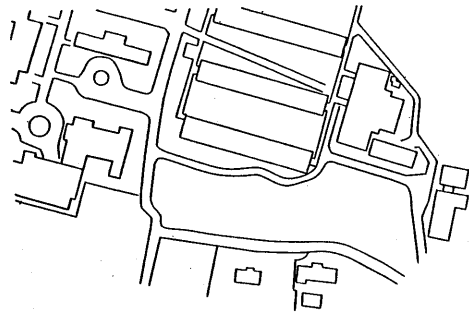
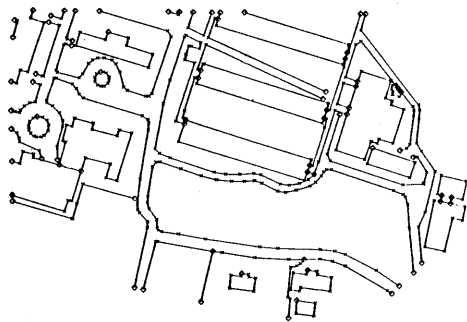


図6 ベクトル化の方法

細線化は画像メモリに入力された図面データに対して、擬似5×5パターンのフィルタリングハー



(a) 入力図面



(b) ベクトル化結果  
○: 端点      ×: 屈折点

図7 ベクトル化結果の例

ドウェアを用いてフィルタリングを繰り返し適用することにより行う。ベクトル化は、ハードウェア/ファームウェアの連携により実行される。まず、細線化された図面データ上で端点を見付け、その点から線を追跡してゆく。細線化された線は8連結であるから、注目点の8近傍を調べることにより容易に端点、交点、屈折点等を検出できる。ベクトル化ユニットを利用して、線を追跡し、結果として端点、交点、屈折点の座標データを得る。図6に示すように斜めの直線では線の途中に多くの屈折点が生じる。また曲線の場合も必要以上に屈折点が発生する。これらの屈折点をそのままの形で保存し、ベクトル化を行うと、ベクトルデータを利用する後続の処理の負担が増大する。そこで、図6に示すように直線近似の誤差の閾値(利用者が自由に設定できる)により、不要な屈折点を削除してベクトル化を行う。すなわち、入力図面と出力ベクトルデータの誤差が与えられた閾値よりも小さい

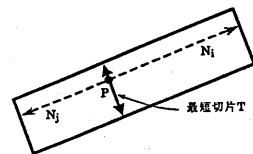
間は、その間の屈折点をすべて削除する。このようにしてベクトル化した図面の例を図7に示す。

この手法では、線を細線化するために線の太さの情報を保存できないこと、及び黒塗り面の形状が失われてしまうことなどの欠点がある。黒塗り面の抽出は、図面全体をある程度細線化してから、輪郭線抽出処理を行うことにより対処している。線の太さをこの手法で抽出することは、図面に種々の太さの線が含まれている場合には、簡単ではない。

## (2) 線を細線化しない手法

線の特徴点の抽出処理は、細線化を行わなくてもできる。例えば、幅のある線を直接追跡し<sup>10)</sup>たり、線の輪郭を追跡する方法<sup>17)</sup>、線の内部の点より触手を伸ばして線の方向を決定する方法<sup>18)</sup>などが現在までに提案されている。

我々は、線の幅は筆点の大きさに依存して決まること、及び線は筆点はその大きさの2倍以上、ある方向に動いた軌跡であるという仮定に基づいて、横断切片検出による特徴点抽出法をホスト計算機上でシミュレーションにより実現した<sup>9)</sup>。横断切片とは、図8に示すように、線分上の任意の点Pにおける最短切片で、その垂直方向の点Pを通るゾンデが両方とも最短切片よりも長いものと定義する。図面の中で横断切片が検出されたところを線領域、検出できなかったところを特徴点領域としてグラフ構造を作成する。図9にこの手法により抽出された特徴点の例を示す。この手法では、線の太さの情報が横断切片の長さとして得られること、線の方向が線上的任意の点で定義できることなどの利点がある。しかし、線が複雑に交差している箇所はほとんどが特徴点となったり、特徴点領域が線を完全に分離しない場合などが問題となる。

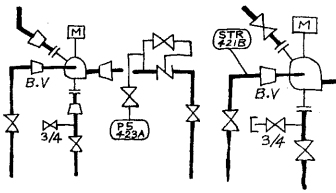


$N_1, N_2$ : 最短切片Tと直交する2つのゾンデ

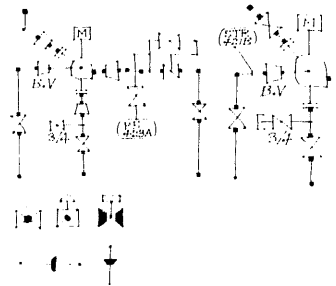
←→ 切片  
- - - ゾンデ

図8 横断切片の定義

グラフ構造作成のための処理は、デジタル画像としての図面の記述をグラフ構造記述に変換するもの



(a) 入力図面



(b) 抽出された特徴点

図9 抽出された特徴点の例

である。グラフ構造記述は図面をベクトル(直線)の集合として記述する。ここでの処理は、図面のもっている有効な情報はそのままの形で保存することが最も重要である。例えば、手書きの線は、それが描かれるスピード・筆圧などにより微妙に太さの違いがあるが、有効な情報としては同一の太さの線であると見なしてよい。ところが、グラフ構造記述では曲線が折れ線近似される。このことにより、有効な情報が落とされたかどうかは後の処理の方式に依存する。この意味で我々が定義したグラフ構造記述は、完全なものではない。

### 3.3 認識・理解処理

グラフ構造作成処理は、対象が線から構成されている画像すべてに適用できるかなり汎用的なボトムアップ処理であった。その結果として作成されたグラフ構造記述をどのように解釈してゆくかは、図面の認識・理解処理の重要な部分であり、トップダウン処理となる。

一般的には、ここでの処理は図10に示すようにさらに細かく分類できる。手書き図面特有の処理は、手

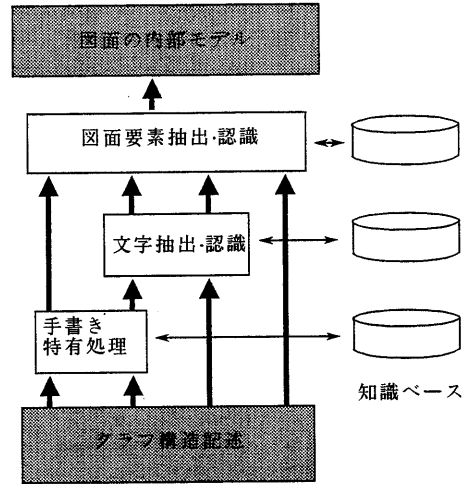


図10 認識・理解処理

書きの線図形によく発生する交点付近の雑音(交点付近での線の途切れ、ヒゲなど)を特に処理する部分である<sup>10)</sup>。この処理を適用すべきかどうかに関する情報がこれまでの処理から得られることが望ましい。文字の抽出・認識処理も図面の種類にかかわらず共通して利用できる可能性が高い。文字認識は英数字の場合はベクトル化処理後でも可能であるが、漢字の場合はベクトル化が余程うまく行かないと難しい。また、線と文字の接触、文字の回転など、簡単な処理では文字の抽出・認識は困難である。

図面要素の抽出・認識処理は、対象に依存する。対象が論理型図面か形状記述図面かにより、図面要素も処理に対する考え方も異なる。理想的には、これら対象に依存する知識を知識ベース化してシステムを構成すべきであるが、我々は、まず、手書きの論理型図面を対象とした処理を実現した。

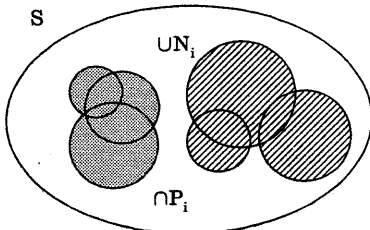
論理型図面は、図面シンボル・接続線・文字をその構成要素とする。これらの構成要素は性質が異なるので、分離する処理が必要である<sup>11)</sup>。図面シンボルは図面の種類により多種多様<sup>12)</sup>であるが、一般的にはループからなるものが多い。接続線は水平・垂直に描かれることがほとんどで、その中には、十分長いものも多く含まれている。文字は、孤立したものが多い。これらのことを考慮して、我々は、特徴点付近の線分の方向性をもとに線分をマージした後、ループ構造を図面シンボルとし、ある程度長い水平・垂直線を接続線とする。次に、これらに接続している線分を順次、判定し

てゆくことにより連結する成分をすべて図面シンボル・接続線に分離する。残ったものを文字とし、文字・図面シンボルに対しては認識処理を施す。

図面シンボル認識は、図面シンボルのもつ幾何学的特徴に基づいて行う<sup>13)</sup>。ある特徴毎に図11に示すように、認識対象シンボルを正・負シンボル集合に分割する。未知のシンボルからどのような特徴が抽出されるかによって、集合演算を行い認識候補を絞る。その後必要ならば、他の特徴を使って詳細認識を行う。化学プラント・論理回路・電気回路図面に用いられるシンボルについて認識実験を行ったところ、90%以上の認識率が得られた。



正負シンボル集合の概念



図面シンボル認識の手順

$NP_i$  : 認識候補シンボル  
 $UN_i$  : 認識候補外シンボル

図11 正・負シンボル集合を用いた図面シンボル認識

文字・図面シンボルの切り出しと認識は、密接な関係にある。すなわち、シンボルの切り出しにはシンボルがどのようなものであるか、という知識が必要である。ところが、シンボルを認識するためには、シンボルがある程度切り出されていることが望ましい。我々は、認識においてリジェクトされ、候補シンボルがなくなると切り出し処理にバックトラックする処理方針でシステムを構成し、実験を進めている。

論理型図面では、この結果のモデル記述として接続関係グラフ記述を生成する。これは、グラフ構造記述

で図面シンボルを構成している線分を1まとめにしたもので、それを予め定められた符号に変換したものである。

形状記述図面では、文字の抽出、寸法線の抽出などが中心となり、それらに基づいた記述方式が考えられる。地図では道路・河川・家屋等の抽出が重要であり、それらに適したモデル記述が考えられる。

このように考えると、認識・理解処理の前半部、すなわち、手書き特有の処理及び文字認識処理は、図面内容の意味的制限を受けないという意味で汎用的である。文字認識、特に漢字の場合はデジタル画像上で行う方が有利なこと、及び文字を含むすべての図面をベクトル化することの是非を考えると、文字認識処理までを一連の処理とし、このレベルで記述形式を設定するのも一手法である。

#### 4. 図面の記述形式と整理処理

##### 4.1 図面の記述形式

FACORESで用いた図面の記述形式は、(1)デジタル画像、(2)グラフ構造記述、(3)図面の内部記述の3種類である。

##### (1) デジタル画像

現実世界の図面をデジタル化したもので、 $m \times n$ 画素( $m, n$ は正整数)の配列、各画素、白(0)、黒(1)の二値画像である。ホスト計算機上では、処理の効率化のために1画素1バイトとして表現しているが、図面処理装置では1画素1ビットで表現しても高速に処理できる。この点が専用ハードウェア装置の特徴である。

##### (2) グラフ構造記述

画像中の黒画素から、線の交差点、屈折点、及び黒塗り面状部分を特徴点として抽出し、特徴点をノード、線分をアークとするグラフで図面を記述するものである。シンボル抽出・認識処理はこの記述に基づいて行われる。グラフ構造の解析を効率的に行うために、ノードとアークは2重リンクで結ばれており、さらに特徴点抽出で得られる情報をその属性として保持する。ノードについては、位置・大きさ・形状があり、アークについては、長さ・方向・線幅がその属性となる。

##### (3) 図面の内部記述

グラフ構造は、シンボル抽出処理によって、図面シンボル・文字列・接続線分を単位として抽出、分

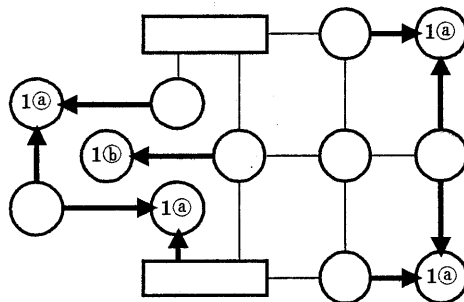
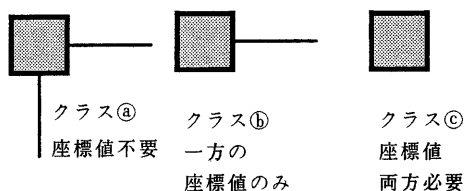


類され、さらに認識処理によって、ノードとアークによる記述が図面シンボル・文字列・接続線分を単位とする記述に置き換えられる。このようにして得られるのが図面の内部記述である。ここで、文字列は、文字コード、大きさ、向き、それが説明するシンボルあるいは接続線へのリンク及び相対的位置関係を属性として持つ。図面シンボルはシンボルコード、大きさ、向き、中心座標を、接続線分は、両端点の座標、線幅、線種(実線、点線)、接続する図面シンボルへのリンクを属性とする。図面の内部記述は、図面に関して得られたすべての情報を保持しているが、冗長な部分もある。よって、本記述を、CAD、論理回路シミュレーション、整列処理など種々の目的に利用することができる。また情報量圧縮、自動索引付けを行い、データベースを構築することもできる。

#### 4.2 整列処理

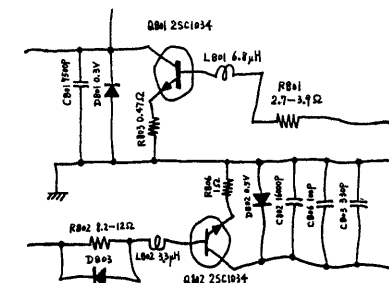
対象が手書き図面である場合、一般的には認識過程から得られる位置情報は、正確ではない。特に、論理型図面の場合、線の形状は重要でないので、図面シンボル・文字・線の特徴点などを整列させることは、図面のモデル記述におけるデータ量の圧縮だけでなく、その記述より生成される図面の整列処理という観点からも重要である<sup>14)</sup>。

図面のモデル記述は、それから原図面と対応のとれた図面を生成できなければならないので、各構成要素及び線の特徴点(これらをまとめてシンボルと呼ぶ)の位置情報を含まなければならない。整列処理とは、各シンボルに配置に関する優先順位を与え、この優先順位の高いシンボルをみの座標値を残し、低いものは高いものに揃えるという処理である。具体的には、図12に示すようにシンボルを分類し、接続線分が水平又は垂直ならば、シンボルと接続線分をペアにして外側から順次、取り除いてゆくことにより、繰り返し処理で優先順位を決定することができる。この優先順位に従って、シンボルを順次配置してゆけばシンボル間の水平・垂直線が折れ曲がることはない。直接接続していないシンボルやシンボル間の間隔などを揃える処理を同時に行えば、整列処理ができる。図13にその結果を示す。この処理は、線の方向性の限定ができれば、拡張は容易であるので、図3に示したようにグラフ構造記述上でも適用できる。この場合は論理型図面でもよい。

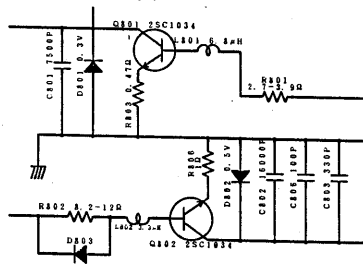


配置に関する優先順位が最も低いシンボルとそのクラス

図12 整列処理のためのシンボル分類と処理手順



(a) 入力図面



(b) 整列結果

図13 整列処理の例

## 5. FACORESの出力処理部

### 5.1 グラフィックス処理

グラフィックス処理は、計算機の内部モデルとしての図面の記述を図面という人間の把握に適したもの

に変換する処理であり、2つの部分に分けられる。前半は図面シンボル展開部で図面シンボルに割当てられた符号より対応するシンボルを生成する処理である。図面の記述には、そのシンボルの大きさ、向き、配置すべき位置情報が含まれている。従って、この部分の処理は単にシンボル辞書(認識するために用いたものでもよい)を参照して、図面の内部モデルに指定された通りにシンボルを生成するだけである。

グラフィックス処理の後半部は、線の両端点をグラフ構造記述より指定され、その間に線を引く処理である。線の太さ、種類、曲線の形状なども指定される。ここでの処理は、既存のグラフィックシステム(CORE, GKS等)<sup>15)</sup>が利用できるので簡単である。

## 5.2 出力時処理

出力時処理は計算機内部のデジタル画像をCRTディスプレイやハードコピー上に出力する時に行う処理である。一般的に考えると、ここでの処理は、計算機内部のデジタル画像と出力結果が同程度の情報量をもっていること、及び出力装置の特性を吸収する処理である。図面の場合、二値画像であるから、解像度の問題だけであり、特別な処理をする必要はない。

## 6. おわりに

本報告では、手書き図面清書システムFACORESについて述べた。手書きの論理回路図の清書処理は、デモンストレーションとして10~15分で人間による対話修正を含めて実現できる。現在は、手書き処理部の構築、形状記述型図面の記述形式の研究、及び文字認識をデジタル画像上で行うことの良否について研究を進めている。

今後は、図面のデータベース化とその検索方式の研究、及び各処理部での種々の方式の比較検討・評価をしてゆく予定である。

[謝辞] 坂井研究室でこの5年間に図面に関する研究に従事した下辻成佳君、林行剛君、大垣武史君、高木宣明君、荒木誠君に感謝します。また、システム作成に御協力いただいた(株)キャディックスの四柳照義部長、木村房夫氏に感謝します。

### [参考文献]

- [1] “手書き設計図面自動入力技術ワークショップ報告書”, 情報処理振興事業協会技術センター (1985)

- [2] T. Pavlidis: "Algorithms for graphics and image processing", Rockville, MD: Computer Science, pp.137-142(1982)
- [3] 岡田至弘、樋野匡利、坂井利之、東田剛志、大田友一: “尾根点・谷点方式による文書画像の前処理付入力装置”, 電子通信学会技術報告EC82-5, p.45-54(1982)
- [4] 王大成、美濃導彦、坂井利之: “3×3単位メッシュ内の濃淡パターンに基づく文書画像領域分割方式”, 画像電子学会誌, Vol.13, No.1 pp.29-37(1984)
- [5] 岡崎洋、美濃導彦、坂井利之: “局所濃淡パターンを利用した線図形の抽出と二値化”, 情報処理学会コンピュータビジョン研究会資料40-2 (1986)
- [6] 荒川賢一、岡田至弘、坂井利之: “濃淡構造情報を利用したカラー文書画像処理”, 情報処理学会第30回全国大会講演論文集, 5N-1 pp.1287-1288 (1985)
- [7] 美濃導彦、坂井利之: “二値線図形に対する画質判定と雑音除去法”, 情報処理学会論文誌 Vol.24, No.2, pp.182-190 (1983)
- [8] 大垣武史、美濃導彦、坂井利之: “市街地図からの白地図の抽出”, 情報処理学会第29回全国大会講演論文集, 6M-7 pp.1145-1146 (1985)
- [9] 林行剛、美濃導彦、坂井利之: “バッファメモリを用いた手書き図面からの特徴点抽出法”, 電子通信学会技術報告PRL85-24, p.43-50 (1985)
- [10] T.Sato, A.Tojo: “Recognition and Understanding of Hand-Drawn Diagrams”, Proc. of 6th ICPR, pp674-677 (1982)
- [11] 林行剛、下辻成佳、美濃導彦、坂井利之: “特徴パターン情報検出による図面の効率的自動入力法について”, 電子通信学会技術報告PRL83-8, pp.57-64 (1983)
- [12] 日本規格協会編: “JISハンドブック図記号”, 日本規格協会 (1982)
- [13] 高木宣明、美濃導彦、坂井利之: “正・負シンボル集合を用いた論理型図面の図面シンボル認識”, 電子通信学会技術報告PRL85-82, pp.49-56 (1986)
- [14] 荒木誠、美濃導彦、坂井利之: “論理型図面の清書方式とデータ量圧縮”, 電子通信学会技術報告PRL85-4, pp.29-37 (1985)
- [15] “Status Report of the Graphic Standards Planning Committee of ACM/SIGGRAPH”, Computer Graphics, Vol. 11, No.3 (1977)
- [16] 上田修功、名倉正計、森克巳: “多値読み取り図面の画質改善手法の検討”, 電子通信学会技術報告IE84-94, pp.49-55 (1984)
- [17] 大沢裕、坂内正夫: “多次元データ構造を用いた図面処理——図面のベクトル化——”, 電子通信学会論文誌 Vol. J68-D, No.4. pp.813-820 (1985)
- [18] 角本繁、宮武孝文、嶋田茂、江尻正員: “実時間色コード化技術を用いた多色図面の自動認識”, 電子通信学会論文誌 Vol.J68-D, No.4. pp.829-836 (1985)