

## 自由作画線図形の折線近似方式

### A Polygonal Approximation Method for Freehand Line Drawings

井上 彰  
Akira Inoue

西川 克彦  
Katsuhiko Nishikawa

岩田 清  
Kiyoshi Iwata

株式会社 富士通研究所  
(FUJITSU LABORATORIES LTD.)

Abstract - This paper describes a polygonal approximation method for freehand line drawings. This method aims at approximating to original patterns as close as possible, and has two features. One is that it does not use thinning which causes an original pattern to be distorted. The other is that it has a hierarchical structure by which the degree of approximation can be changed depending on the complexity of line drawings to realize an efficient processing. We applied this method to city maps and confirmed that it can obtain high accurate approximation results.

#### 1 まえがき

近年、計算機利用の高度化に伴って、各種設計図面の自動入力の研究が積極的に行われ、一部実用レベルに達しつつある<sup>(1)</sup>。これらのシステムで取り扱われる図面は、各種製造CADの入力の対象となり、定型化され規約に沿って書かれたものである。

ところで、地図、イラスト等の線図形に着目してみるとその形状にはなんら制約のない自由に作画されたものである。これらの線図形を計算機に入力し、蓄積加工する場合には、可能な限り原面に忠実でしかもデータ圧縮率を上げるという相反する要求に応える技術の開発が必要となる。本発表はこれらの技術の一つとして開発した自由作画折線近似方式に関するものである。

筆者らは、これまでにプリント板パターン図や論理回路図のような一定の格子に作画された図面の処理技術について提案<sup>(2)</sup><sup>(3)</sup>してきたが、地図等の自由作画図面を原面に忠実に表現できる折線近似方式はこうした技術を土台として開発したものである。以下、本稿では開発した折線近似方式の基本的な考え方、処理方式および実験結果について述べる。

#### 2 基本的な考え方

図面の折線近似技術としては従来、細線化処理を基にした追跡法<sup>(4)</sup>などが提案されているが、これらの手法は細線化による画像の劣化のため近似精度に難があるとか線図形の近似が画一的になり適応的な近似ができないという問題がある。こうした問題を解決するための一つの手法とし

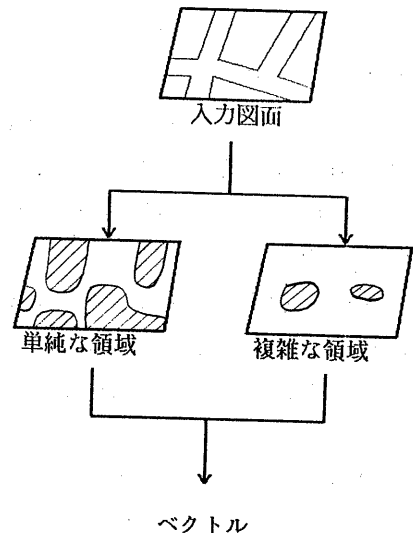


図1 本方式の概念図

て、筆者らは、線図形の直線性に基づいて図面全体を単純な領域と複雑な領域に分け、各々に対して適応的にベクトル化近似を行うことによって、原画に忠実でかつ処理効率のよい折線近似方式を開発した。この考えに基づいて提案する本手法の概念を図1に示す。

### 3 処理方式

#### 3.1 基本構成

図2は2章で述べた基本的な考え方に基づいて実現した折線近似方式の基本構成である。本方式では線図形の直線性に着目して直線的な部分と、交点、変曲点などの非直線部分に分け、直線的な部分は粗い処理、非直線的な部分はより細かな処理を行うことによって処理効率の向上を狙ったものである。処理の流れは、領域分類、ベクトル化およびベクトル統合の3段階の処理からなっている。

領域分類では、線図形の複雑さに応じて処理の細かさを変化させ効率的に折線近似を行うために、図面内の線図形の直線らしさを尺度として図面全体を単純な領域と複雑な領域とに分類する。領域分類を一括的に処理すると、膨大な処理を要するとともに領域の判定基準が難しくなるので本方式では矩形単位に直線/非直線を判定し、それをまとめることによって領域分類の簡易化と高速化を目指している。具体的にはまず矩形単位に分割し、各矩形毎に直線/非直線を判定し、連続した同一属性の矩形をまとめることによって領域を分類する。最終的に、単純な領域はその構成線図形は直線（以後“直線領域”と呼ぶ）となり、一方複雑な領域は線図形の交点、変曲点等を含む部分（以後“非直線領域”と呼ぶ）となる。

ベクトル化では、領域分類で区分された二つの領域に対し、適応的にベクトル化処理を行い、複雑な部分は原画に忠実になるよう精密な近似を、単純な部分は粗い近似で処理効率の向上を目指している。直線領域のベクトル化は領域分類で抽出した特徴だけを用いた数値処理となり、非直線領域では領域分類で抽出した特徴だけでは不十分であるため画像から輪郭形状に着目したさらに細かな特徴を抽出するというイメージ処理となる。

ベクトル統合では、2つの領域のそれぞれで得られたベクトルを統合して、図面全体の折線近似を得る。

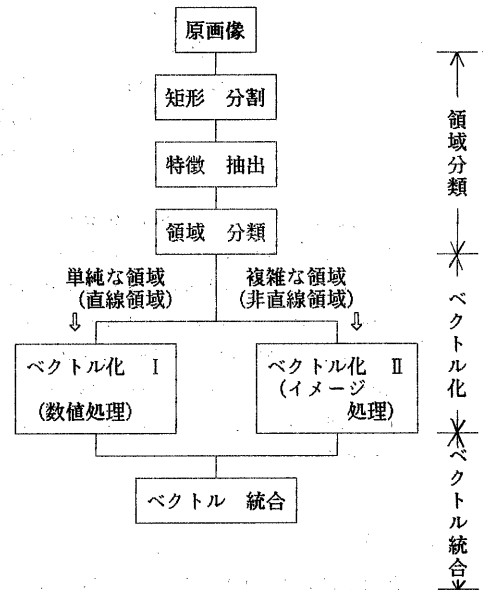


図2 本方式の基本構成

#### 3.2 領域分類

入力図面を直線領域と非直線領域に分類する処理は、矩形分割、特徴抽出および領域分類の3段階からなり、以下各々について説明する。

##### 3.2.1 矩形分割

原画像全体をある一定間隔の格子で分割する。図3は対象とした市街地図の原画像の例(9.45ドット/mm)であり、図4はそれを一定間隔(WDドット)の格子(ここでは32×32ドット)で分割したものである。これは原画像の情報圧縮の一環として行うもので、図形全体を直線領域と非直線領域に分類するための最小単位となるものである。

##### 3.2.2 特徴抽出

ここでは、3.2.1の矩形分割で得られた各々の矩形が直線か非直線かを判定するための特徴抽出を行う。具体的には各々の矩形内の図形にラベル付けをほどこし、ラベル付けされたものに対し、連続性を示す特徴として格子の上アドレスおよび線図形の直線性を示す特徴として方向性ヒスト

グラムを抽出する。上記2つの特徴はいずれも線図形の輪郭線から抽出するものであるため、ラベル付けは線図形の輪郭線に対して行うだけで十分である。

図5は矩形内の原画像の1例であり、図6は図5に対して輪郭線のラベル付けを施したものである。

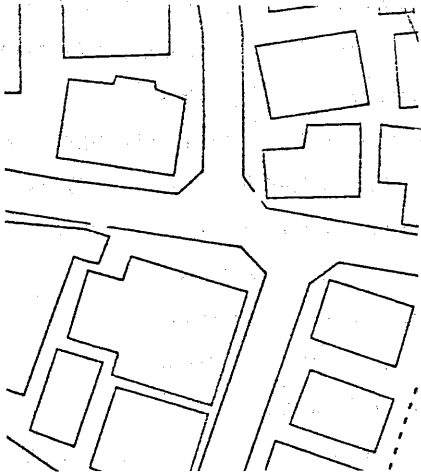


図3 原画像

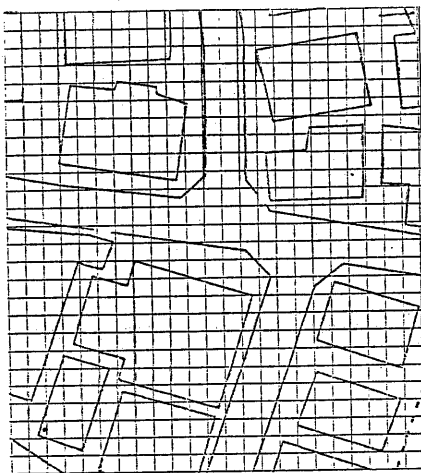


図4 格子により分割された原画像

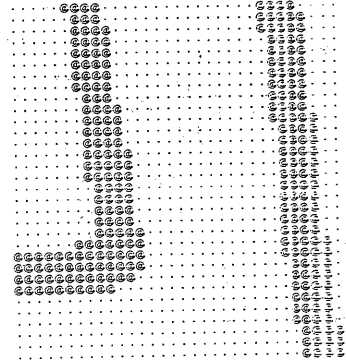


図5 矩形内の原画像の例

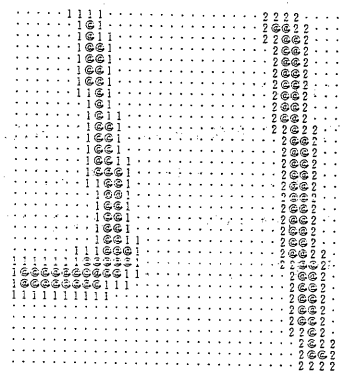


図6 輪郭線のラベル付け

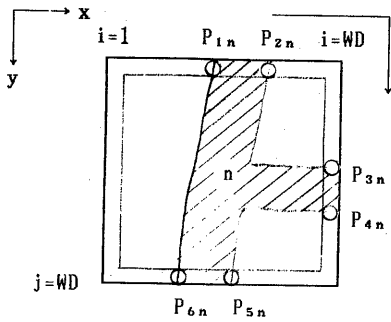
抽出する特徴は以下の通りである。

① 格子上アドレス

格子上に存在する線図形の輪郭線上の点（以後“接点”と呼ぶ）のアドレスを抽出する。図7において、矩形の境界の部分（ $i=1$  or  $WD, j=1$  or  $WD$ ）で、矢印の向きに  $n \rightarrow 0$  または  $0 \rightarrow n$ （ $n$ はラベル）と変化する点のアドレス  $P_{kn}(x_{kn}, y_{kn})$  およびその個数  $S_n$  を求める。

② 方向性ヒストグラム

線分の直線性を表現する特徴として  $3 \times 3$  画素のウィンドウを用いて線図形の輪郭上の各画素の方向性（8方向）を示す方向性ヒストグラム（HIST(i)）を抽出する。

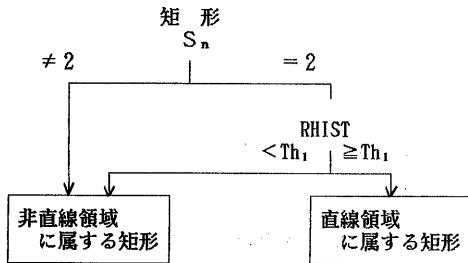


n : 対象とするラベル

図7 格子アドレス

### 3. 2. 3 領域分類

各矩形内の線図形の直線性を判定し、それを直線、非直線の2種類に分類し、さらに隣接する同一属性の矩形を結合して入力図形全体を直線領域と非直線領域に分離するという2段階の処理からなっている。



ここで  
 $S_n$  : 接点の数

RHIST :

$$RHIST=MAX \left[ \frac{\sum_{i=kk}^{kk} HIST(i)}{\sum_{i=1}^8 HIST(i)} \right]$$

$i=1,8$   $kk; i=1\sim7$  のとき  $kk=k+1$   
 $i=8$  のとき  $kk=1$

図8 矩形の分類

### (1) 矩形の分類

矩形を3.2.2で述べた特徴量に基づいて、図8に示す条件によって直線部分と非直線部分に区分する。この処理をラベルnごとに繰り返す。

### (2) 矩形の結合

同一属性の矩形が連続しているか否かを格子上アドレスを基に判断し、連続している矩形の結合を行う。具体的な結合手順は図9に示すように、矩形の座標(i,j)を中心として(i,j-1)と(i-1,j)を以下の手順で結合する。

#### ・矩形(i,j)と(i,j-1)との結合

矩形(i,j)と(i,j-1)との境界線上に接点が存在するとき、それぞれの接点を  $P_{1c}(x_{1c}, y_{1c})$ ,  $P_{2c}(x_{2c}, y_{2c})$  および  $P_{1u}(x_{1u}, y_{1u})$ ,  $P_{2u}(x_{2u}, y_{2u})$  とすると

$$x_{1c} \leq x_{2u} \text{ かつ } x_{1u} \leq x_{2c} \text{ のとき}$$

矩形(i,j)と(i,j-1)を結合する。

#### ・矩形(i,j)と(i-1,j)との結合

上記の場合と同様に、

$$y_{4c} \leq y_{2L} \text{ かつ } y_{1L} \leq y_{3c} \text{ のとき}$$

矩形(i,j)と(i-1,j)を結合する。

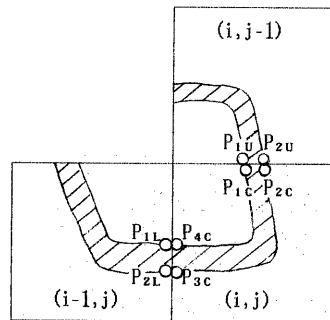


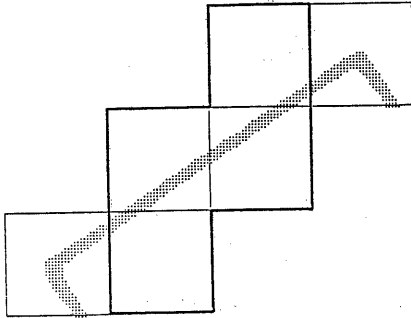
図9 矩形の結合

図10は直線領域の例であり、図11は非直線領域の例である。

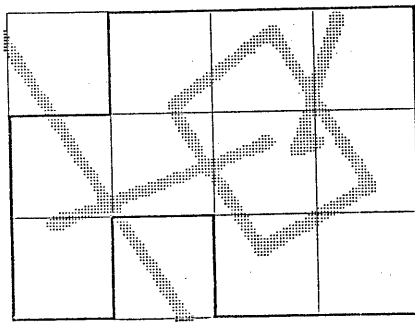
### 3. 3 ベクトル化

領域分類された直線領域と非直線領域に対して、それぞれベクトル化I, IIを行う。ベクトル化Iは矩形内は交点、

変曲点等を持たないので格子アドレスの情報だけでベクトル化が可能となり、これらの数値処理で実現できる。一方、ベクトル化Ⅱは矩形内に交点、変曲点等を持つため格子アドレスだけでは情報不足となり、画像から線図形の輪郭形状を細かく調べるというイメージ処理を行う。



太枠は直線領域を示す  
図10 直線領域の例



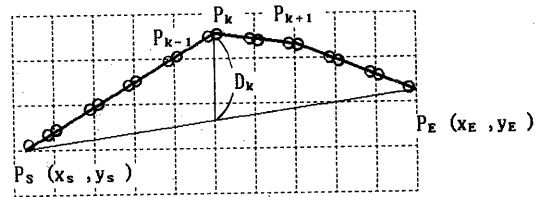
太枠は非直線領域を示す  
図11 非直線領域の例

### 3. 3. 1 ベクトル化Ⅰ

図12は処理手順の説明図である。図12で○印で示した点が前述の特徴抽出で得られた接点である。本処理では、特徴抽出で得られた接点のうち、真の変曲点になるものを選択する。具体的には、 $\text{Max}(D_k)$  が近似のための閾値 $\text{Th}_2$  以上のとき、点  $P_k$  を変曲点として抽出する。

ここで

$$D_k = \begin{cases} |y_k - \{(y_E - y_S)(x_k - x_S) / (x_E - x_S) + y_S\}| \\ : |x_E - x_S| \geq |y_E - y_S| \text{ のとき} \\ |x_k - \{(x_E - x_S)(y_k - y_S) / (y_E - y_S) + x_S\}| \\ : |x_E - x_S| < |y_E - y_S| \text{ のとき} \end{cases}$$



$P_S(x_S, y_S)$ ,  $P_E(x_E, y_E)$  は直線領域の両端点

図12 ベクトル化Ⅰ処理手順の説明図

このようにして抽出された変曲点  $P_k$  を新たな接点と考え、上記の処理を $\text{Max}(D_k) < \text{Th}_2$  となるまで繰り返すことにより最終的なベクトルが得られる。

### 3. 3. 2 ベクトル化Ⅱ

本処理は、まず線図形の輪郭線のベクトル化を行い、その後、対となる輪郭線ベクトルを統合して、線図形のベクトル化を実現するものである。

#### (1) 輪郭線のベクトル化

ここでは線図形の輪郭線の局所的な方向性を表現する傾きコードを基にして、その傾きコードの隣接状態を調べ、同じ傾きコードが連続する部分を1つの輪郭線ベクトルとして抽出する。

#### (2) 輪郭線ベクトルの統合

(1)で抽出した輪郭線ベクトルの対(線図形の両側の輪郭線を表現する)を求める。具体的には、相互の距離

(D), 重なり度 (P) を次式のように定義する。P は 0 から 1 の間の値をとり, 1 に近いほど重なり度が高いことを意味する。したがって P が 1 に近いものの中で D が最小のものを対ベクトルと決定する (図13参照)。

$$D = (D_1 + D_2) / 2$$

$$P = K'_2 - K'_1$$

$$\left[ \begin{array}{l} K'_1 = \max \{ 0, K_1 \} \\ K'_2 = \min \{ 1, K_2 \} \\ K_1 = \frac{\vec{P}_3 \vec{P}'_1}{\vec{P}_3 \vec{P}_4} \\ K_2 = \frac{\vec{P}_3 \vec{P}_2}{\vec{P}_3 \vec{P}_4} \end{array} \right]$$

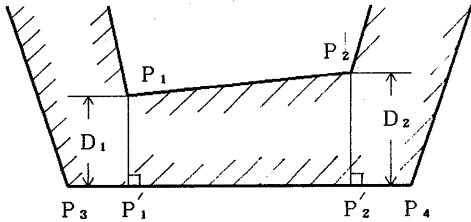


図13 輪郭線ベクトルの統合

次に対となる輪郭線ベクトルを統合してベクトル近似を行うが, 次の2つの統合の処理が代表的である。

・交差のない場合 (図14)

輪郭線ベクトル  $L_1, L_2$  が対となり, 屈曲点  $V_1, V_2$  を統合し, その中心から変曲点  $P_1$  を求めることにより輪郭線ベクトルを統合する。図14に示すように, 各対ベクトルが統合されて, 最終的なベクトル化 (破線部) が行われる。

・交差のある場合 (図15)

まず, 輪郭線ベクトル  $L_4, L_5$  が対となることから  $V_1, V_2$  が対の屈曲点として抽出される。一方, 対の輪郭線ベクトル  $L_1, L_3$  に対して,  $L_3$  上に屈曲点  $V_1$  に対応する仮の屈曲点  $D_1$  を生成する。同様に  $L_2, L_3$  に対して,  $V_2$  に対応する仮の屈曲点

$D_2$  が生成される。これらの対輪郭線ベクトルの関係から  $V_1, V_2, D_1, D_2$  の4点は1点に統合され, その重心  $P_1$  が最終的な変曲点となり, 図中破線のようにベクトル化が実施される。

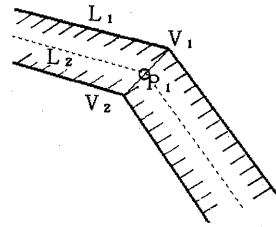


図14 輪郭線ベクトルの統合手順 (交差のない場合)

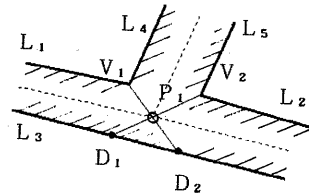


図15 輪郭線ベクトルの統合手順 (交差のある場合)

3. 4 ベクトル統合

ここではベクトル化 I および II で得られた直線領域および非直線領域それぞれのベクトルを両領域の境界上で統合を行う。すなわち両領域の境界で連続しているベクトルを1本のベクトルに統合できるかどうかを判断し, ベクトル化された2つの領域の統合を行う。本処理は, 原理的にはベクトル化 I で用いた判定基準と同じものを用いるため, その詳細は省略する。

#### 4. 実験結果

図16は実際の市街地図の例である。図面サイズはA2サイズである(入力に用いたドラムスキャナの解像度は9.45ドット/mmである)。図17はこの市街地図の一部を拡大したものであり、図18はその処理結果である。また、近似精度を見るために市街地図のごく一部を拡大して従来方式<sup>(4)</sup>と比較したものが図19~21である。原画像(図19)に対して、本方式を用いてベクトル化を行った結果が図20であり細線化処理を用いた従来方式でベクトル化を行った結果が図21である。本方式は、従来方式と比較して図形の交点、変曲点部分で改善がなされており精度の高い近似結果が得られていることが分かる。

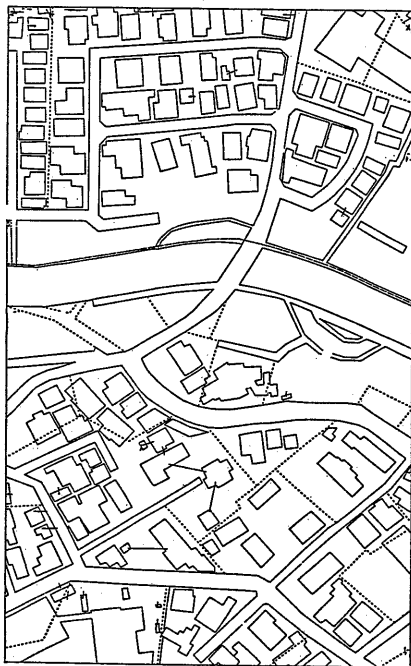


図16 市街地図の例

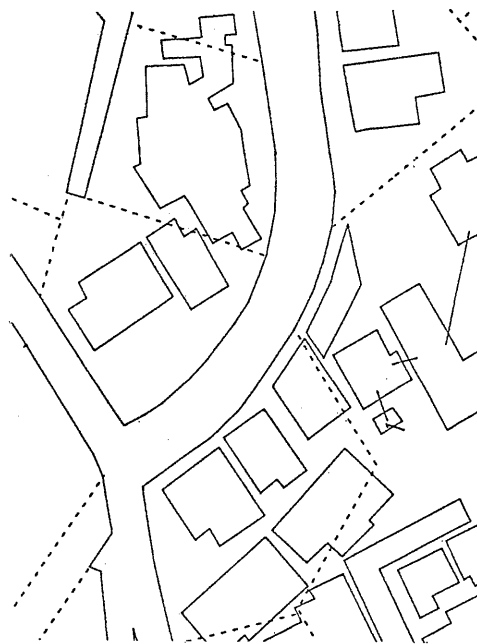


図17 市街地図の一部の拡大図

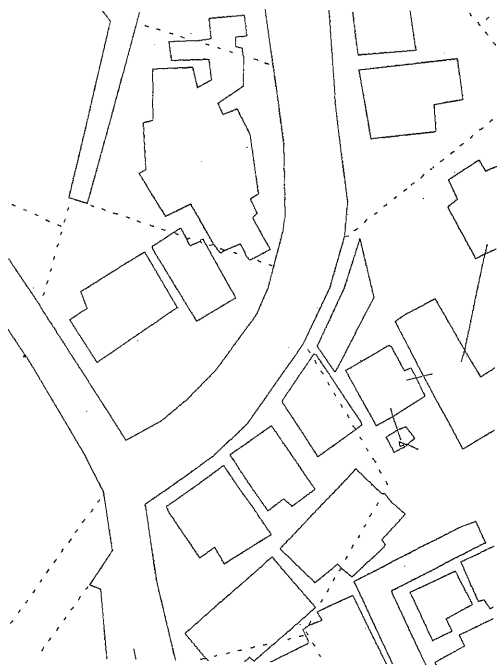


図18 処理結果

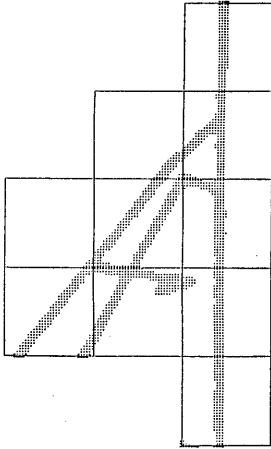


図19 市街地図のごく一部の拡大図

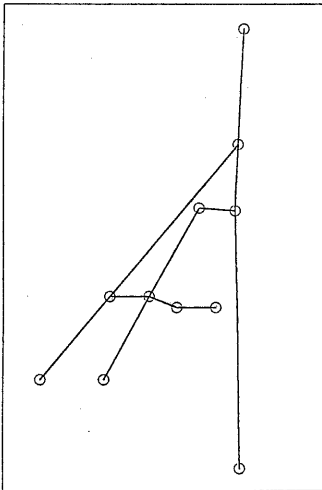


図20 本方式の処理結果

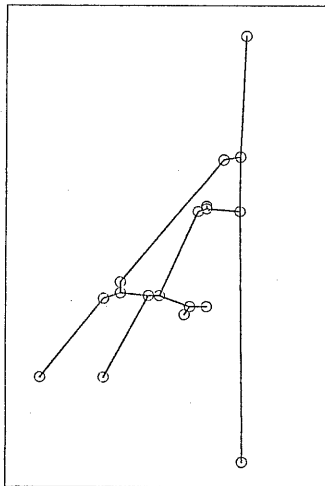


図21 従来方式の処理結果

## 5. まとめ

本稿では、自由に作画された線図形を原画に忠実に折線近似する方式について述べた。本処理方式は以下のような特長を持つ。

### (1) 近似精度がよい

細線化処理を用いない構造解析的な処理による近似方式であるため、細線化処理から派生する線分の縮退、交点の変形、ヒゲの発生等の影響による近似精度の低下がない。

### (2) 階層的処理構成

図面全体を矩形小領域に分割し内部構造が複雑な部分についてだけ高次の処理を用いることにより効率的な処理ができる。

本方式を用いることにより比較的高精度の処理結果が得られることが確認された。今後は、各種の図面を対象として本方式の評価・改良を行う予定である。

## 謝辞

日頃御指導いただく峯島部長および関係各位に深謝する。

## 参考文献

- (1) “特集1 図面読取りの最新技術”, PIXEL, No. 21, pp62-107 (1984).
- (2) 清水, 榊井他 “設計図面の自動入力/処理装置”, 電子通信学会研究会資料, PRL 80-15, pp97-103 (1980).
- (3) 長田, 井上他 “線画像認識方式—文字と図形の混在する図面”, 情報処理学会研究会資料, CV 17-3 (1982).
- (4) 長田, 吉田他 “線図形の記述方式”, 電子通信学会研究会資料, PRL 79-50, pp27-36 (1979).