

スプライン関数による図面の修復

小林 富士男*・今城 勳*†・坪井 始*・田中 始男*・美咲 隆吉*

The Restoration of Drawing by Spline Function

Fujio KOBAYASHI, Isao IMAJYOU, Hajime TUBOI,
Motoo TANAKA and Takayoshi MISAKI

ABSTRACT

Recently, there are many studies dealing with automatic input of drawings to a computer. In drawings, lines are overlapped by characters or symbols in many places. Extraction of characters and restoration of drawings after elimination of characters require laborious jobs. In this paper, an automatic restoration method of drawings is proposed. This method reduces largely labor required for restoration. First, algorithms for various processes are described. Next, experimental result of application of this method to architectural drawing is shown. Lastly, we consider several examples of the results of this paper.

キーワード：図面、修復アルゴリズム、スプライン関数、補間、文字の削除

Key Words : drawing, restoration algorism, spline function, interpolation, elimination of character

1. まえがき

一般に図面中には、線図形の他に多数の文字・記号などが存在し、それらの文字・記号を図面中から削除すると、線図形に重畳した箇所では、必要な線図形も同時に削除され分断がおこる。その修復作業を人手によると、多大な時間と労力を必要とする上に誤差も生じる。そこで、修復作業を計算機によって自動的に行うことが望まれている^{(1)~(5)}。

本研究では、文字・記号などの削除によって分断された線図形の修復を目的としている。分断された線図形の接続手法としては、与えられた点（離散点）を確実に通る3次のスプライン関数を採用し、図面に適用する際のアルゴリズムを開発した^{(6),(7)}。本論文では、開発したアルゴリズムについて述べ、さらに、実際の建築用図面に適用して得られた処理結果を示す。また、その有効性を確認している。

2. 修復処理

本章では、分断によって生じた端点の探索、対応端点の選出、端点間の接続処理などについて述べる。さらに、補間するために採用する離散点を求める際に考慮しなければならない事項とそれらに対する処理方法について記述する。図1に、その処理の流れ図を示す。なお、本論分で述べる主要な処理は、原図面から求めた細線化図面に対して行う。

2.1 入力図面

対象図面をイメージスキャナによって読み取り、得られた256階調の図面を2値化したものを原図面とする。さらに、線追跡処理などの簡便化のため、原図面から細線化図面を生成し、その両図面を処理対象とする。両図面共に、図形部分は黒画素、背景部分は白画素である。

2.2 文字・記号などの削除 (図1の☆1)

図面中に存在する文字・記号などの削除は、マウスを用いて削除領域を指定し、その指定された領域内の黒画素を削除する。図2は、線図形の交差部分と文字が重畳している例であるが、その削除領域を点線枠で示す。

2.3 補間候補点の探索 (図1の☆2)

指定した削除領域に従って文字・記号などを削除すると、線図形も同時に削除され、線図形が分断される。細線化図面を対象に、分断によって生じた端点の探索を行う。なお、端点の探索を行う領域は、図3に示すように、

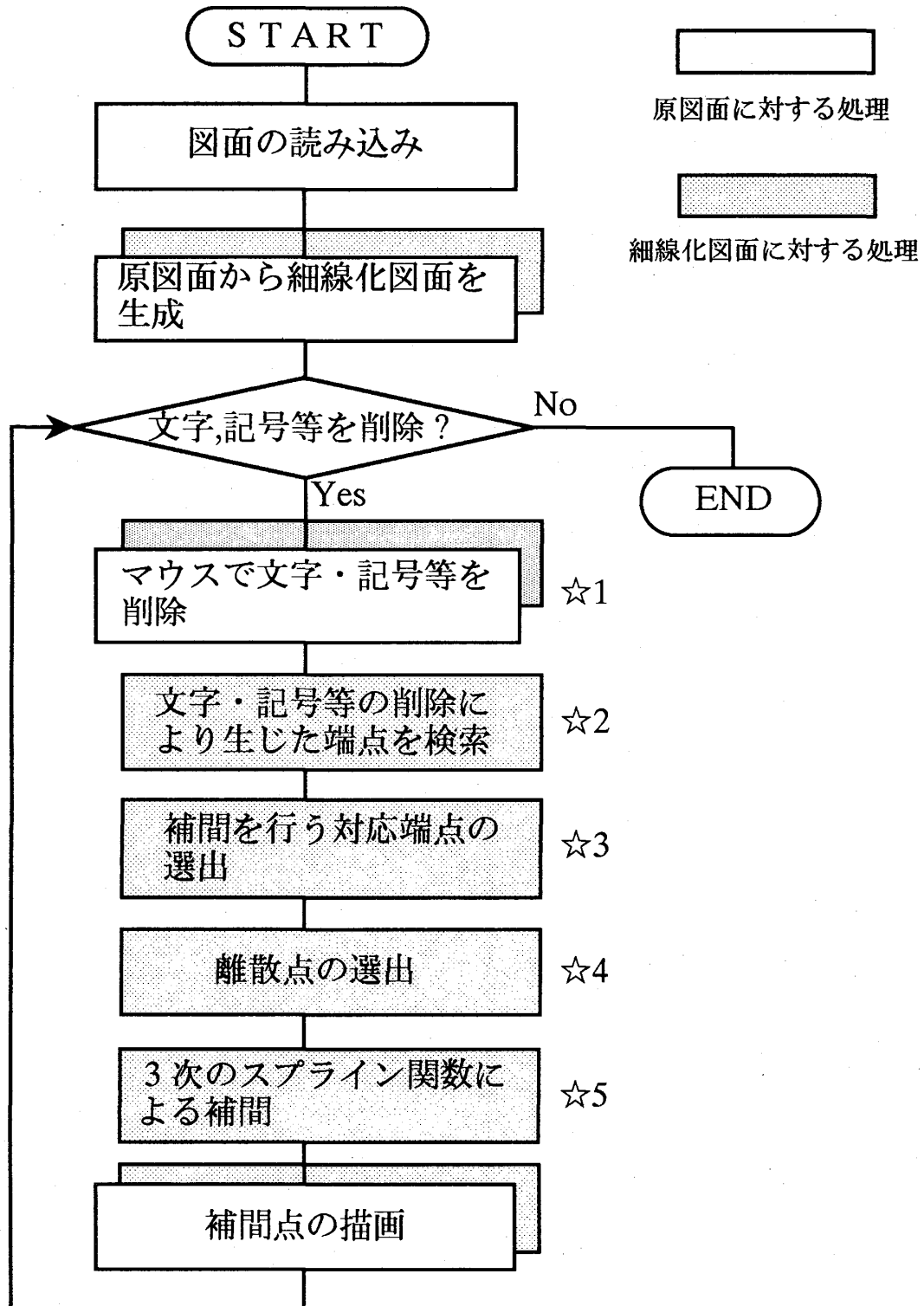


図1 処理の流れ図

文字・記号などの削除領域より W だけ拡張した領域とする。ただし、 W は数画素である。

探索領域を走査し、黒画素に遭遇すれば8近傍を調べる。その場合、連結している画素が1画素であれば、分断によって生じた端点とみなし補間候補端点 EP_i ($i=1,$

$2, \dots$)として記録する。さらに、 P_i を端点にもつ線図形をそれぞれ L_i とする。図4に分断された線図形と補間候補端点を示す。

2.4 対応端点の選出 (図1の☆3)

補間候補端点の中で、互いに接続を行う対応端点の選

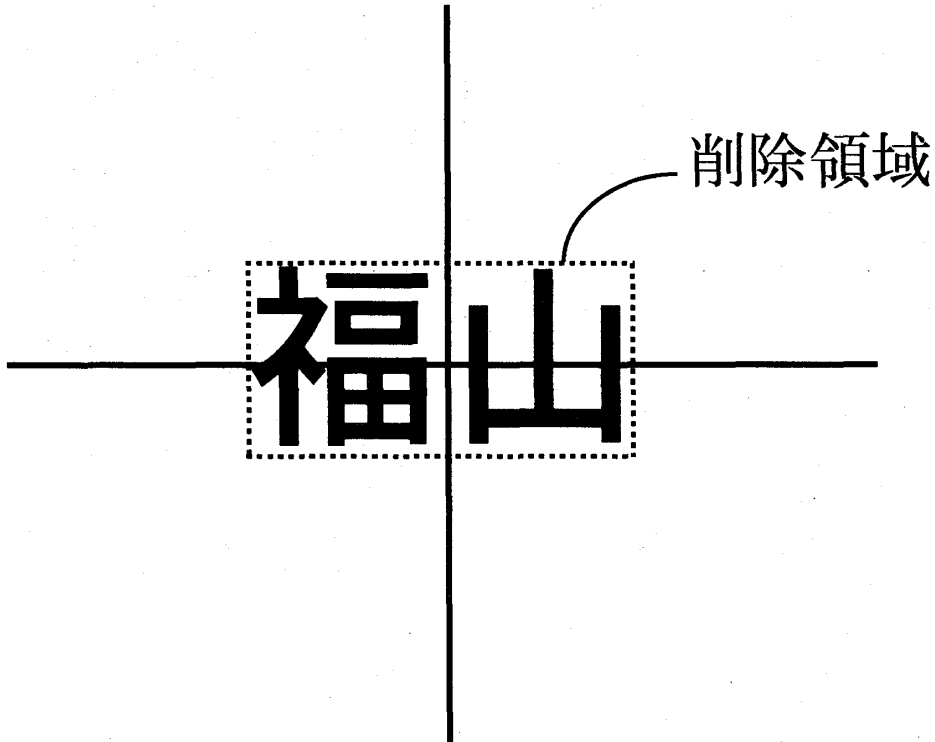


図2 線図形の交差部に重畳した文字

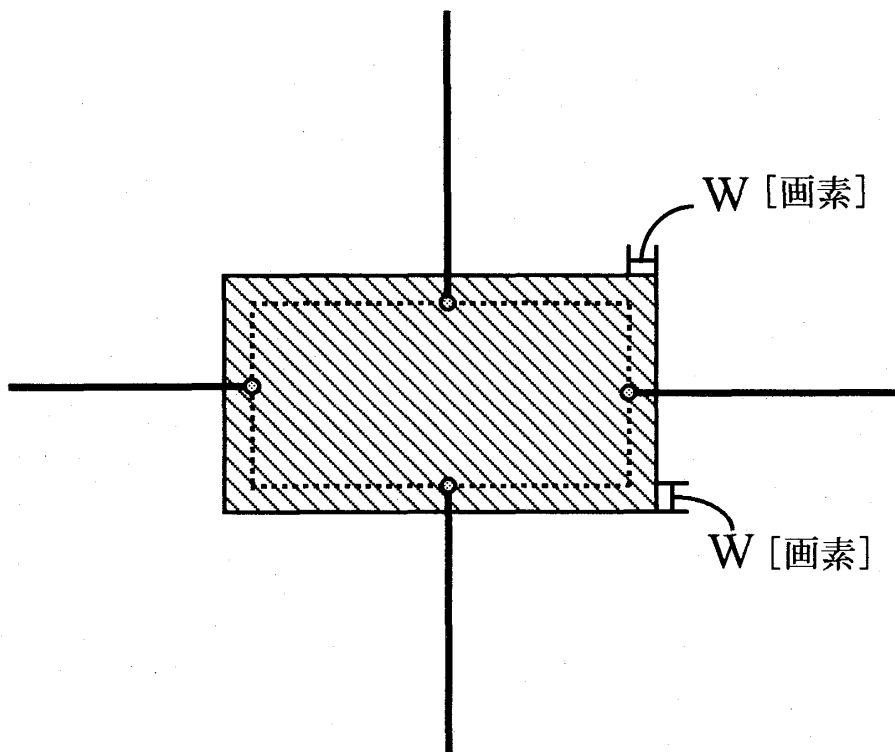


図3 端点を探索する領域

出処理を細線化図面に対して行う。図5 ($i=1$ の場合)に示すように、分断された線図形 L_1 上の点から端点 EP_i に向かうベクトルを VM_i ($i=1, 2, \dots$)とし、そのベクトルの大きさを LA とする。ただし、 LA は数画素である。一方、 EP_i 以外の端点 EP_j ($j=1, 2, \dots$)

から、逆に線図形 L_j 上に向かうベクトルを VS_j とし、その大きさを VM_i と同じにする。ここで、 $i \neq j$ である。さらに、端点 EP_i から端点 EP_j に向かうベクトルを $VP_{i,j}$ とする。いま求めたベクトルを用いて、端点 EP_i の対応端点を選出する時の処理を次に述べる (図6参照)。

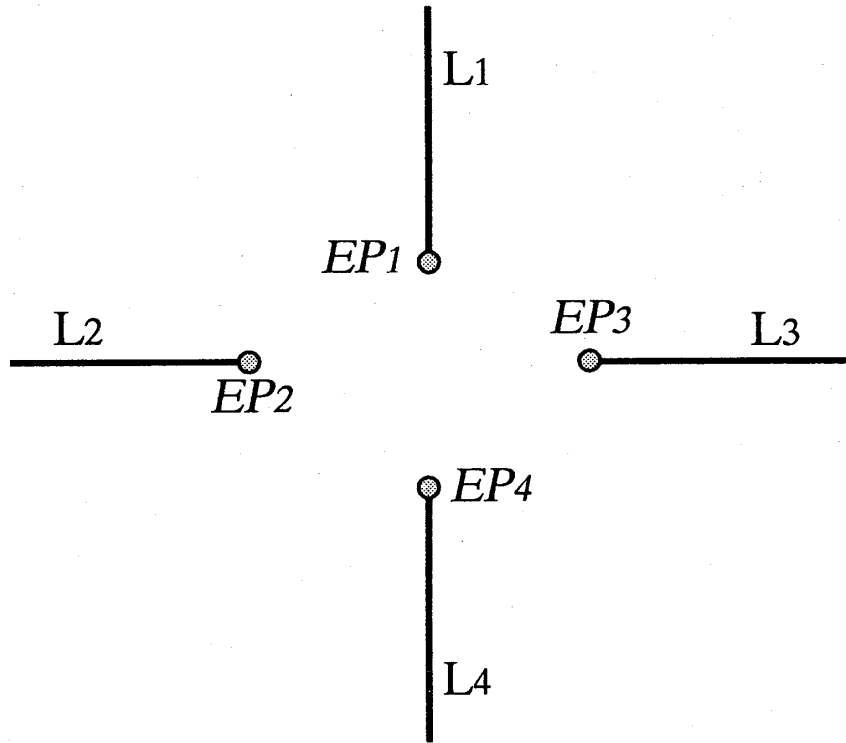


図4 分断された線図形と補間候補端点

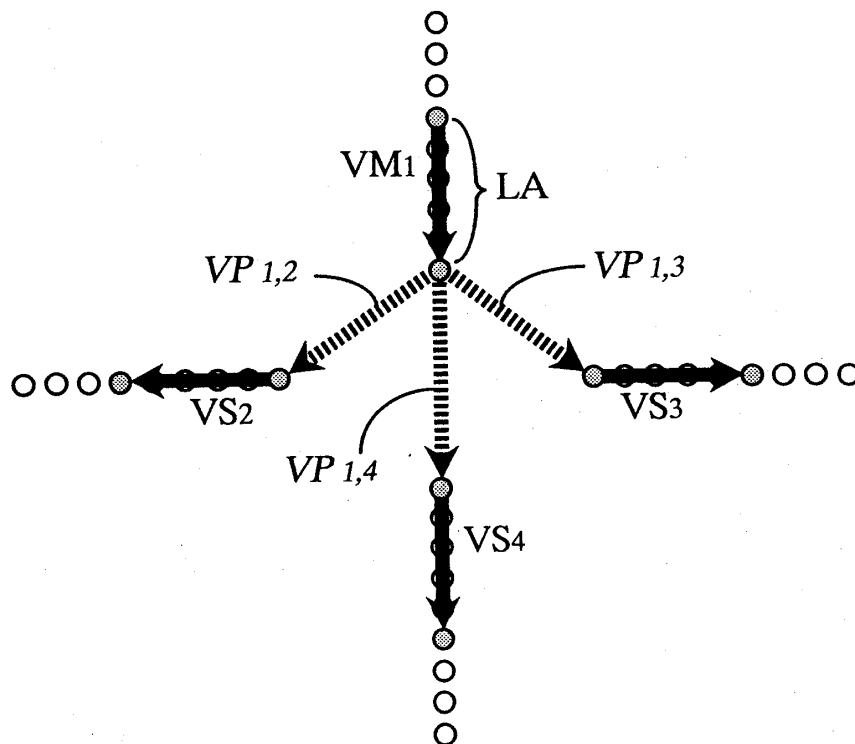


図5 対応端点の選出時に考慮するベクトル

処理 (I) : ベクトル $VP_{i,j}$ を基準とし、 $VP_{i,j}$ と VM_i および VS_j との方向差が、次の式(1)、(2)を同時に満足するならば、端点 EP_i と EP_j を対応端点として記録する。ただし、 θ_1 は閾値、($i=1, 2, \dots$; $j=1, 2, \dots$) である。

$$|VP_{i,j} - VM_i| < \theta_1 \quad (1)$$

$$|VP_{i,j} - VS_j| < \theta_1 \quad (2)$$

上記の処理を i, j の全ての値について繰り返し実行する。なお、式(1)、(2)を同時に満足する端点が存在しない場合は、対応端点なしとする。

図5の場合には、処理(I)により EP_1 と EP_4 が対応端点として選出される。

2.5 離散点の選出 (図1の☆4)

対応端点が求めれば、3次のスプライン関数により端点間の接続を行うわけであるが、その際に採用する離散点を選出しなければならない。そこで、接続を行う対応端点の近傍の線図形上に離散点候補を求め、その候補の中から実際に採用する離散点を選出する。本節では、離散点の選出方法、考慮すべき事項および条件を述べる。

2.5.1 離散点の選出方法

- (1) 端点をそれぞれ PL, PR とし、 PL, PR 間を結ぶ直線距離を L とする (図7参照)。
- (2) 各々の端点 PL, PR から L だけ離れた点まで線追跡を行い、それらの点を PF, PB とする。

(3) 線追跡を行った PL, PF 間と PR, PB 間の全ての点を離散点候補とする。

(4) いま求めた離散点候補の中から、隣合う点の距離が等しくなる点を離散点として選出する (図8参照)。

2.5.2 考慮すべき事項と条件

(a) 線追跡上に他の線図形が交差している場合

図9に示すように、他の線図形が追跡線上に交差していることがある。この場合、端点からの線追跡途中に線図形の交差部分である分岐点に出会ったならば、その分岐点からさらに線追跡を続ける方向を選択しなければならない。そこで、いままで線追跡を行って線図形上の点から分岐点に向かうベクトルを VB とし、そのベクトルの始点から分岐点までの大きさを LB とする。ただし、 LB は数画素である。さらに、分岐点を始点として、線追跡を行う各線図形上へ向かうベクトルを VT_k ($k=1, 2, \dots$) とし、その各ベクトルの大きさを VB と同じにする。いま求めたベクトル VB と VT_k を用い、分岐点からの追跡方向を選択する処理を次に述べる。

処理 (II) : ベクトル VB を基準とし、その VB と VT_k の方向差が最小のものをを選び、次の式(3)を満足するならば、 VT_k の方向に線追跡を続ける。ただし、 θ_2 は閾値、($k=1, 2, \dots$) である。

$$\min_k (|VB - VT_k|) < \theta_2 \quad (3)$$

なお、式(3)が満足されなければ、分岐点

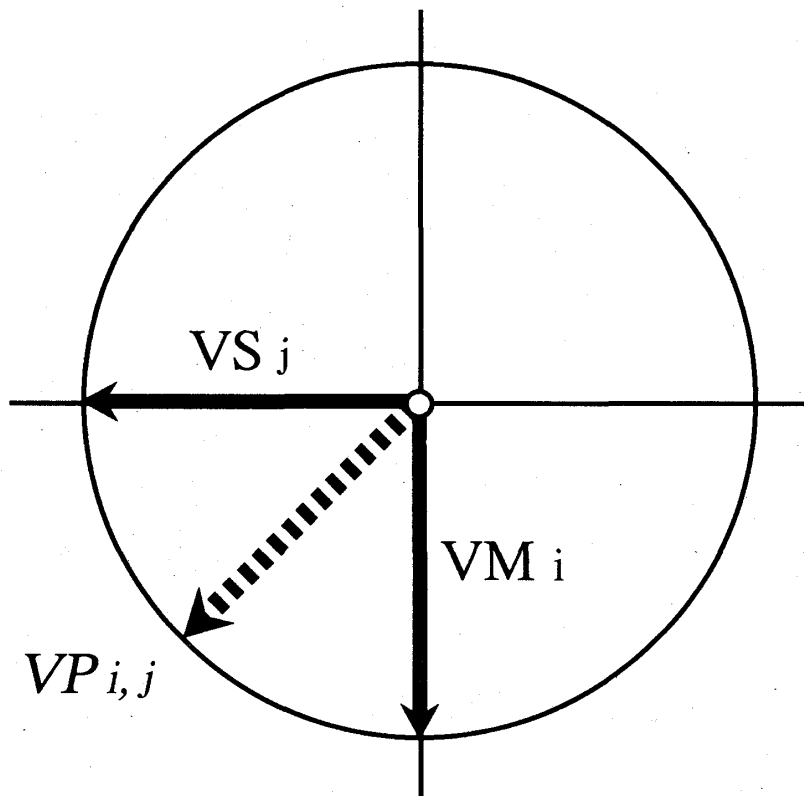


図6 ベクトルの方向差

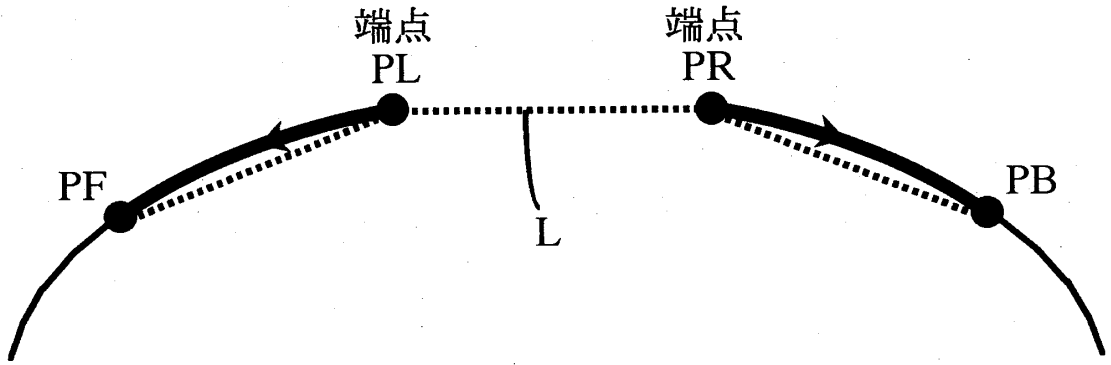


図7 離散点候補を求める範囲

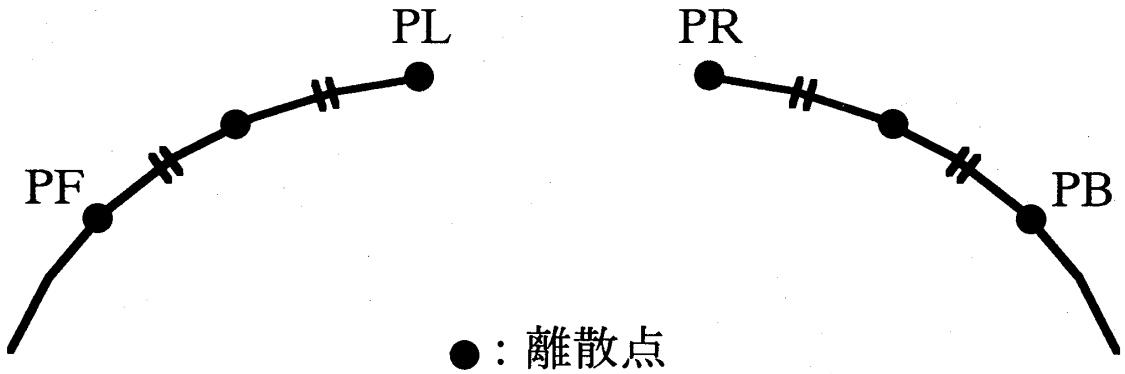


図8 選出した離散点

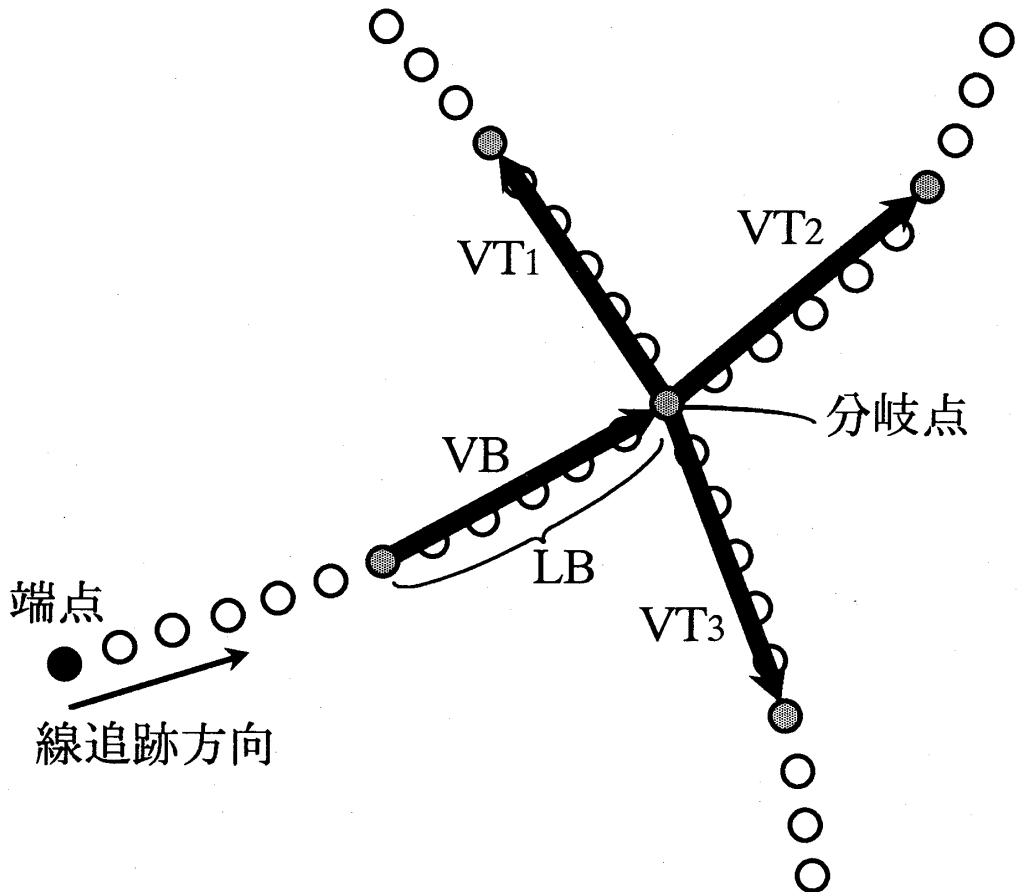


図9 追跡方向の決定に考慮するベクトル

で線追跡を終了し、端点から分岐点までの線追跡された全ての点を離散点候補とする。

図9の場合には、 VT_2 の方向へ線追跡が行われる。

(b) 離散点候補の範囲を限定しなければならない場合

図10は、真円の弧の一部が文字削除により分断された例である。この場合、端点PL、PR間の接続を行うために各端点から両端点間の長さLだけ各々の線上を線追跡すると、離散点候補を求める範囲は、図11の DSB_1 、と DSB_2 になる。しかし、 DSB_2 のように互いに性質の異なる

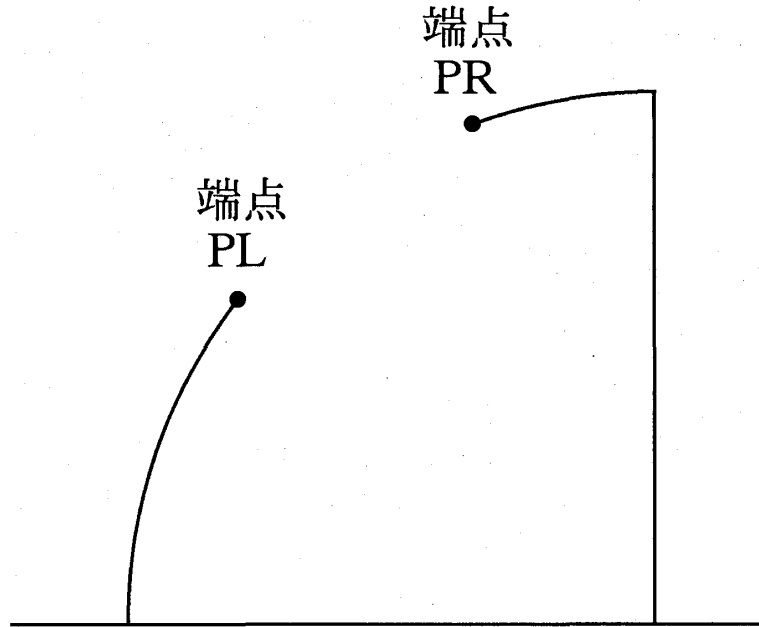


図10 線図形が分断された例

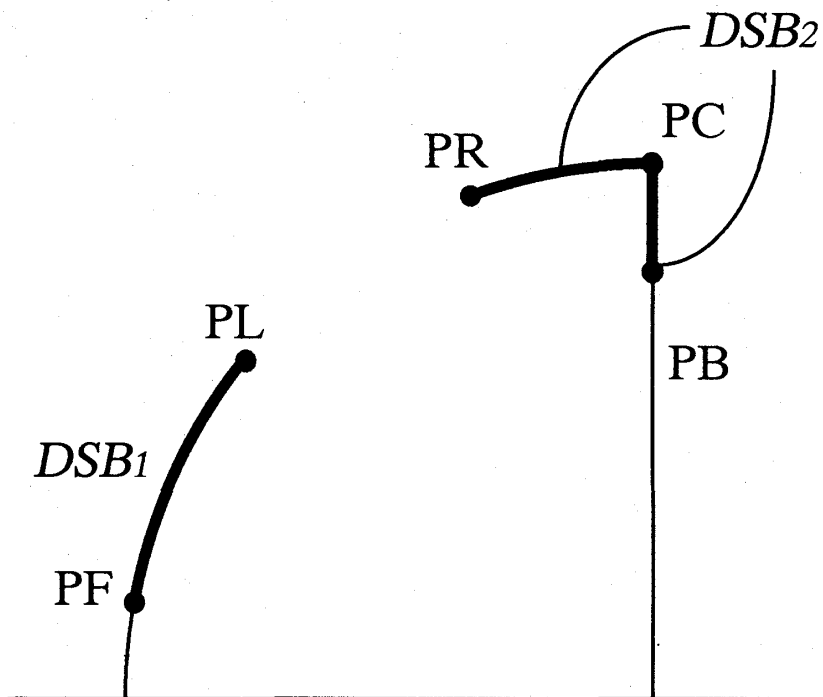


図11 離散点候補を求める範囲

る線図形上に離散点候補を求め、それらの中から離散点を選出すると補間する際に振動が発生しやすくなるので、PR、PC間に DBS_2 を制限することが望ましい。そこで、図12のように対応端点PL、PRを始点として線追跡を行いながら、長さLCを単位としてベクトルを求め、 VL_i 、 VR_i ($i=1, 2, \dots$)とする。ここで、LCは数画素とする。これらのベクトルより、離散点候補の範囲を制限する処理を次に述べる。

処理 (III) : 連続する2つのベクトル VL_i 、 VL_{i+1} および VR_i 、 VR_{i+1} の方向差が、次式を満足するならば、そのまま線追跡を続ける。ただし、 θ_3 は閾値、($i=1, 2, \dots$)である。

$$|VL_i - VL_{i+1}| < \theta_3 \quad (4)$$

$$|VR_i - VR_{i+1}| < \theta_3 \quad (5)$$

なお、式(4)が満足されなくなったとき、端点PLからベクトル VL_i までを離散点候補とし、また、式(5)が満足されなくなったとき、端点PRからベクトル VR_i までを離散点候補とする。

2.6 対応端点間の補間接続 (図1の☆5)

選出した対応端点と離散点より、3次のスプライン関数を用いて端点間を補間接続する。なお、端点間は、1画素の線幅で補間を行っている。

3. 処理結果

本章では、本論分で述べたアルゴリズムを手書きの建築用図面に適用した処理結果を示す。図13(a)は原図面で、縮尺1/100の手書き建築用図面を解像度150 [dpi]、階調8 [bit] で入力し、2値化を行った図面である。その

図面の画素数は、 400×640 である。図13(b)は、原図面から生成した細線化図面である。図14に原図面上で指定した文字の削除領域を示す。図15に、上述したアルゴリズムによって求めた離散点を示す。本処理では、 $W=2$ [画素]、 $LA=LC=5$ [画素]、 $LB=10$ [画素]、 $\theta_1=5$ [deg]、 $\theta_2=\theta_3=20$ [deg] とし、上述の条件を満足する場合には、両端3個ずつ計6個の離散点を採用して分断箇所の補間を行っている。これらの値は、予備実験によって決定した。図16は、本アルゴリズムで処理した修復図面である。

4. 考察

修復図面から分かるように、上述したアルゴリズムによって文字・記号などの削除によって分断された図面の修復処理が可能となった。ここで本手法の考察を行う。

4.1 対応端点の選出処理

原図面中に存在する文字列の多くは直線図形に重畳しており、その直線図形での対応端点の選出は正しく行われている。また、図14の文字列「1,900」のよに一つの削除領域に複数の線図形が重なっている場合でも、対応点が正しく選出され補間されている。

4.2 交差している場合の線追跡処理

図14の中央付近に存在する文字列「3,800」の削除後、端点から線追跡を行い離散点を求める場合が、この処理に該当する。文字列「3,800」の右側に存在する線図形の交差部分において、分岐点から水平方向に線追跡が行われ、離散点はその線上で正しく求まっている。

4.3 離散点候補の範囲を制限する場合の処理

図14の文字列「1,900」の削除後、端点から線追跡を行う場合がこの処理に該当する。各々の端点からの線追跡

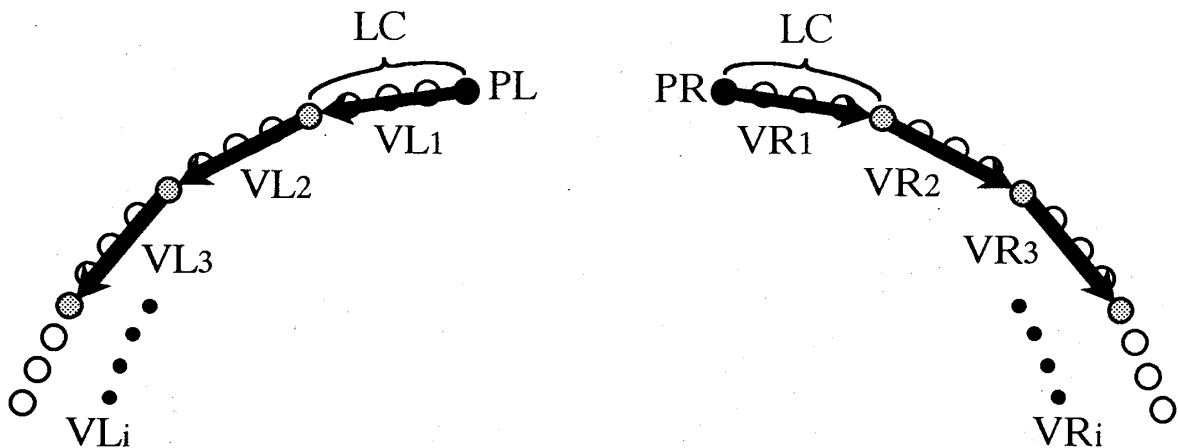
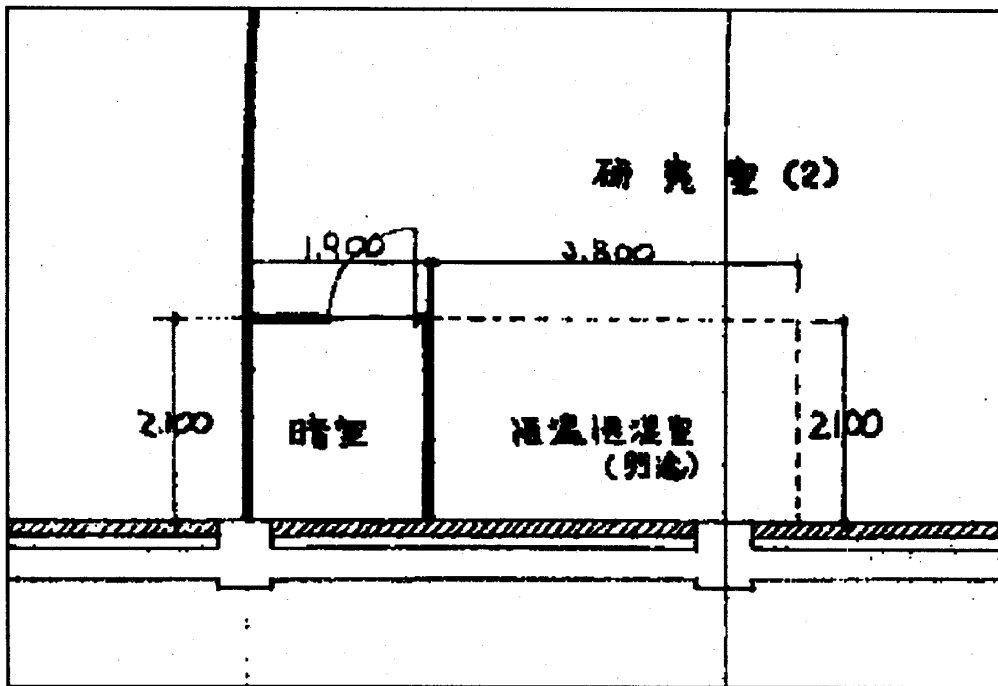
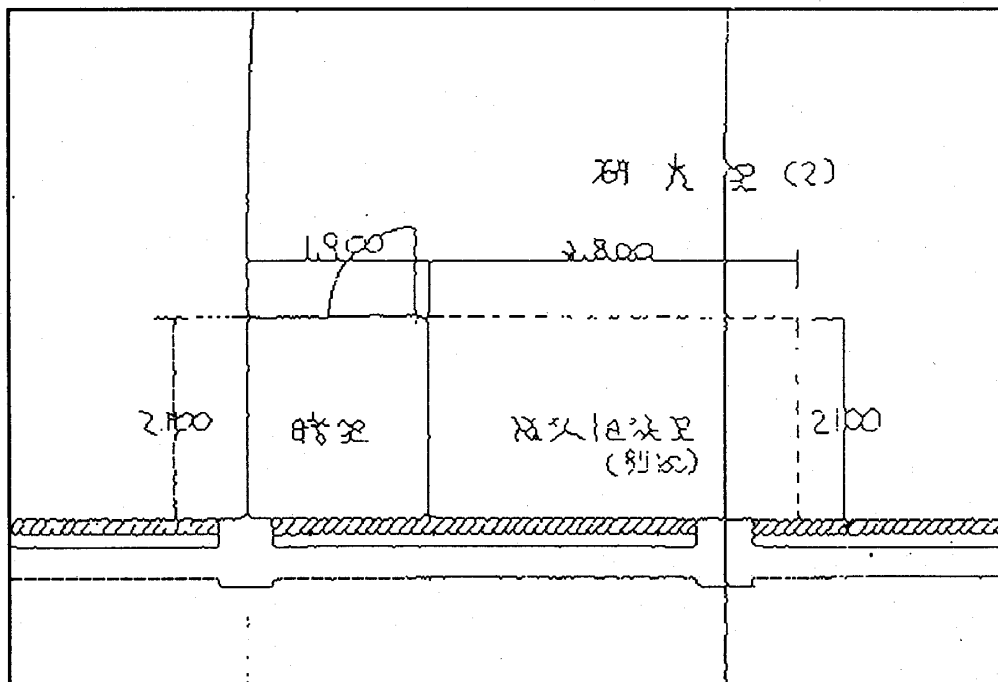


図12 追跡線上に求めたベクトル



(a) 原図面



(b) 細線化図面

図 13 処理対象の図面

処理で、追跡途中に線図形の曲率が大きく変化する箇所において線追跡を終了し、離散点を求める範囲が制限されている。

上述の通り、いずれの処理においても良好な結果が得られ、その有効性が確認された。しかし、補間の際少し

振動することもあった。その原因は、図面のデジタル化や細線化の際に発生する誤差の影響である。

5. むすび

本論分では、文字・記号などの削除によって線図形が

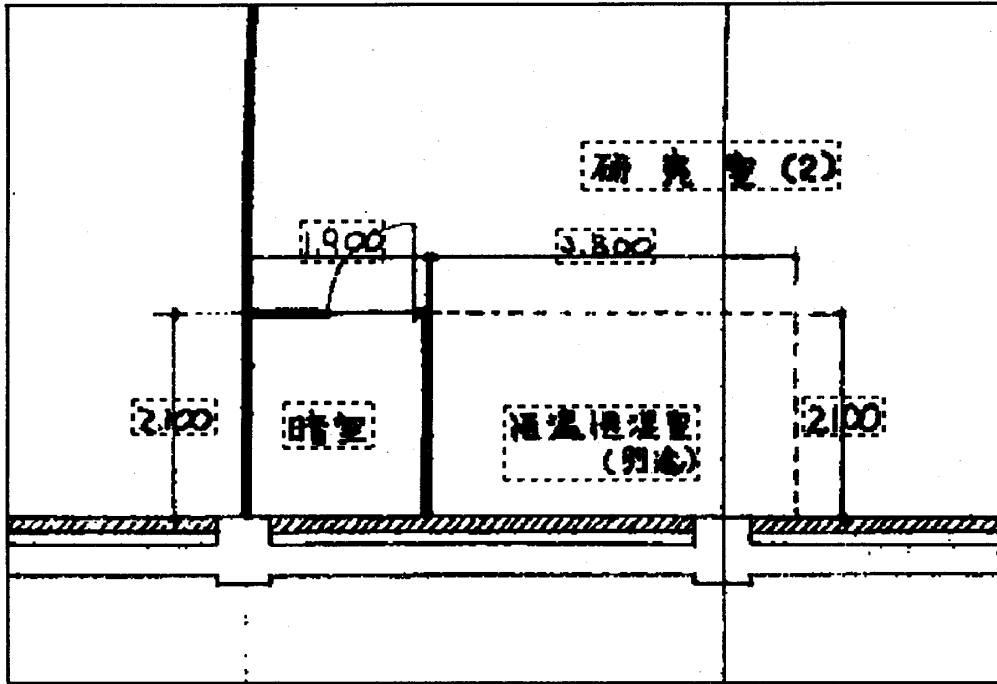


図 14 文字列の削除領域

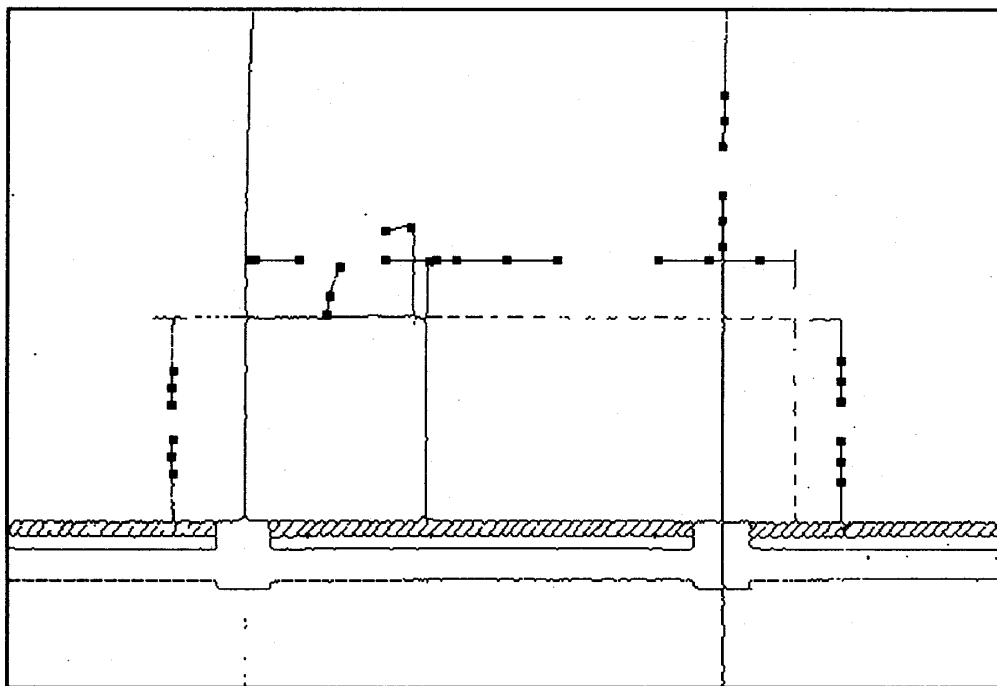


図 15 細線化図面上に求めた離散点 (各補間に 6 点を考慮)

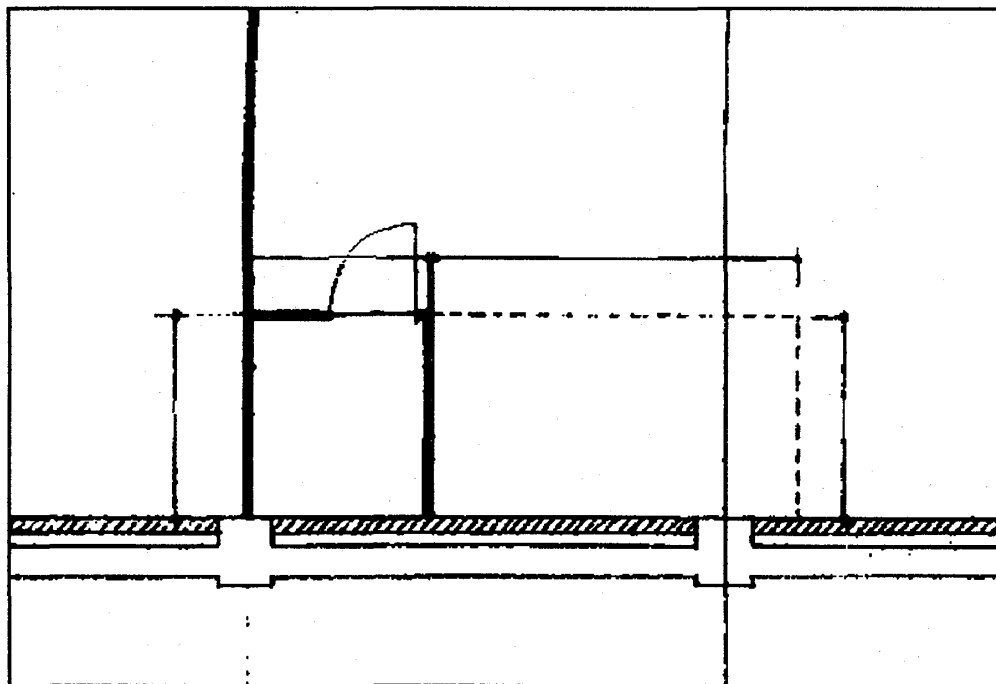


図 16 修復図面

分断された箇所を修復する方法について述べた。開発したアルゴリズムを実際の建築用図面に適用し、その有用性を確認した。現在は文字領域の抽出を人間の介在に依存しているが、これらの処理を自動化するアルゴリズムを開発すれば、大幅な時間と労力の軽減が期待される。今後の課題としては、精度の向上と、原図面の線幅で補間するように改良する事であると考えている。

会中国支部第44回連合大会、No.142021 (1993).

- (7) 小林、今城、坪井、田中、美咲：“スプライン補間法による図面の修復処理”、平成6年電気学会全国大会、No.1747 (1994).

参考文献

- (1) 宮武、松島、江尻：“平行線抽出手法を用いた地図からの道路情報自動抽出”、信学論、Vol.J68-D, No.2, pp.153-160 (1985).
- (2) 大沢、滝嶋、坂内：“会話的な認識による信頼性の向上を図った地図自動入力システム”、信学論、Vol. J72-D-II, No.4, pp.545-554(1989).
- (3) 長尾、安居院、中嶋：“文字列抽出後の2万5千分の1地形図の修復処理”、画像電子学会誌、Vol.19, No.3, pp.131-137(1990).
- (4) 小林、大村、美咲、坪井、田中：“市街地地図の道路網の抽出”、平成3年度電気・情報関連学会中国支部第42回連合大会、No.041904 (1991).
- (5) 小林、大村、田中、坪井、美咲：“市街地図からの道路抽出”、福山大学工学部紀要、第14号、pp.85-90 (1992).
- (6) 小林、今城、坪井、田中、美咲：“線図形の修復処理に関する基礎的検討”、平成5年度電気・情報関連学