書道ロボットに関する研究

—— ひらがな文字の生成 ——

尾家　寛哉*・藤本　真作**・逢坂　一正**

*岡山理科大学大学院工学研究科機械工学専攻修士課程

**岡山理科大学工学部機械工学科

(1996年10月7日　受理)

1. はじめに

我々人間は簡単に文字を書くことができる。そのとき，手本があればそれを目で見る，または記憶の中から探しだし頭の中でどのような文字であるかを認識して中心線を見いだし軌道を生成する。次にその情報を腕に伝え実際に文字を書くという手順をふんでいる。

もしこれをロボットが行うとすれば，カメラを使って手本をデータとして取り込みコンピュータで画像処理する，もしくは記憶装置から手本の情報を読み込み認識する。そして，軌道を生成しその情報をマニピュレータのコントローラに伝え制御して文字を書く，というふうに置き換えることができる。

そして，この一連の手順の中で重要なのは，文字認識問題，軌道生成問題，制御系設計・実装問題である。

ここでは軌道生成の問題について，我々は今まで文字の形は中心線の軌道が重要であり中心線軌道を実現するために最小二乗法による一筆書きひらがなの軌道生成[1)]を行ってきた。しかしながら1）の方法では，一価な座標を持つひらがなでなければ軌道を得ることができず，軌道の座標データ数，最小二乗法の次数ともに非常に大きくしなければならない，という問題があった。また中心線を必ず通るとは限らないので手本の軌道からずれてしまう，という問題もあった。そこで本報告では抽出した中心線データとパラメトリック スプライン関数を使った軌道生成の方法を提案し，中心線データとの誤差を減少させ，軌道生成可能となるひらがなの数を増やすことを目的とした実験結果について述べる。

2. 画像処理[2)]

コンピュータで計算をさせるために必要なデータを抽出するため，もしくは，抽出しやすくするために画像処理を行わなければならない。そのために，

まず，初期画像データは

- 256階調グレイスケールイメージ画像
- 画像サイズ 256×256 pixels
- 解像度　1.7 pixels/㎜

の3つの条件を満たすものとする。そして初期画像データと，あるしきい値の画像濃度の差分をとり正であれば1，負であれば0の値を与え2値化画像を作成する。そして，値1の点の4近傍の微分を行い枠線を抽出する。また，得られた枠線からひらがなの中心線データを求める。

2.1　2 値 化[2)]

2値化はしきい値処理とも呼ばれ，入力画像（初期画像データ）を $F=\{f_{ij}\}$，出力画像を $G=\{g_{ij}\}$ としたとき，パラメータn(しきい値)を用いて(1)式の操作を行うものである。

$$g_{ij}=\begin{cases}1\,;\,f_{ij}\geq n\\0\,;\,f_{ij}<n\end{cases}\tag{1}$$

今回は文字と背景の分離であるためしきい値選択処理は行わず，n = 127をしきい値として固定し2値化処理を行っている。

2.2　枠線抽出（エッジ検出処理）

エッジを検出する場合，ある点とその近傍点の濃度が図1のように急激に変化する（傾きがある一定以上の値を示す）場所をエッジとみなし出力する。

この操作は微分を行うことで容易に求めることができる。

この操作をデジタル的に考えるとある点とその近傍点の濃度差分を取ることに等しい。そこで，画面上のある点 (x, y) とその4近傍各点 $(x+1, y)$, $(x, y-1)$, $(x-1, y)$, $(x, y+1)$ との濃度差分を行う。今回は2値化画像のエッジ検出であるため濃度にかわる値は0か1

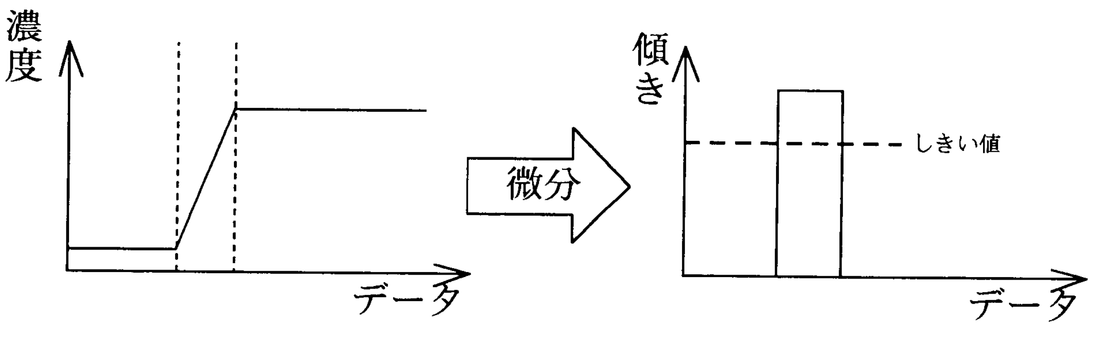


図1　エッジ

しか存在しない。ゆえに，差分が0でない点が4近傍のなかに1つ以上存在する場合に(x, y)を座標データとして残すことによって図2(ii)のように枠線を抽出できる。

2.3 中心線データの作成

中心を求める方法は細線化，骨格化などの方法があるが，これらは画像データを外側から削り落としてゆき，幅1となったときの画像を結果として出力するというものである。

しかし，この操作を軌道の中心線データを求めるために用いたとすると，端点部分が失われてしまい正確な中心線データを抽出することは出来ない。(付録参照)

そこで我々は枠線データを用いた中心線データの抽出方法を提案する。この方法を用いれば端点の中心線データを失うことがなく，その点における幅も求めることができる。

ある幅をもった図2(ii)のような画像の中心線は幅の長さを2等分する点の集まりであるといえる。

ゆえに，枠線を図3(i)のように①枠，②枠と分けたとき①枠のある点から最小距離にある②枠の点を探索し，この最小距離を2等分する座標点を中心線の座標とする。そして①枠からの探索が終了すれば今度は②からも同様の操作を施し図3(ii)の中心線が得られる。

3. 軌道生成[3)]

軌道生成における軌道の補間はラグランジュ多項式などが知られているが，データ数に対して次数があがり，それにともない補間した軌道は振動的になってゆく。そこで我々はデータ数に関係なく次数を低次に設定できるスプライン関数を用いた。そして，スプライン関数の中でも$x-y$平面上の任意の点を表すことのできるパラメトリックスプライン関数を使って軌道生成を行うことを提案する。

3.1 スプライン関数

スプライン関数の特徴は，(2)式のように異なった接点間$(q_i \leq x < q_{i+1})$それぞれに多項式が構成され，図4に示すようにすべての接点を通り，かつ滑らかに接点をつなぐことのできる区分的多項式関数であるということである。

また，補間関数として使用される場合，1つの式で補間関数を表す多項式がデータN個に対して$(N-1)$次の多項式を形成するのに対し，スプライン関数ではデータ数に関係なく次数を区間ごとに任意に設定して補間を行うことができるという利点を持っている。

$$y = \begin{cases} a_0 + a_1x + \dots + a_nx^n \quad (q_0 \leqq x < q_1) \\ b_0 + b_1x + \dots + b_nx^n \quad (q_1 \leqq x < q_2) \\ \quad\vdots \qquad\qquad\qquad \vdots \end{cases} \tag{2}$$

3.2　スプライン関数の表記法

一般にスプライン関数を表す場合(2)式のように区間ごとに多項式を書いて表す。しかしこの表記法ではコンピュータでの計算に不便である。

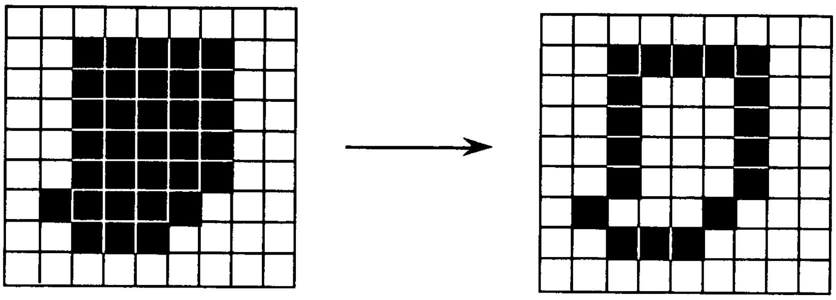
そこで，これをコンピュータによる解析に用いるための方法について以下に示す。

まず，切断すべき関数と $(K-1)$ 次のB—スプライン関数を以下のように定義する。

$$y=(x-q_1)_+^{K-1}=\begin{cases}0 & (x<q_1)\\(x-q_i)^{K-1} & (x\geqq q_1)\end{cases}\tag{3}$$

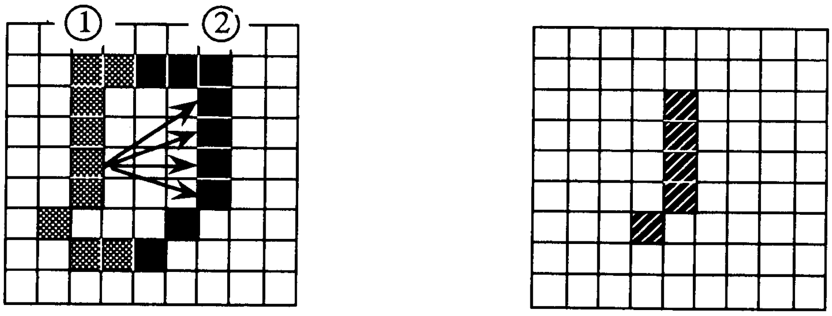
$$B_{i,K}(x)=(-1)^K(q_{i+K}-q_i)\{\frac{(x-q_i)_+^{K-1}}{p_i(q_i)}+\frac{(x-q_{i+1})_+^{K-1}}{p_{i+1}(q_{i+1})}+\ldots+\frac{(x-q_{i+K})_+^{K-1}}{p_{i+K}(q_{i+K})}\}\tag{4}$$

ここで，x, yは座標データ，q_iは接点を表す。



(i)　(ii)

図2　枠線抽出



(i)　(ii)

図3　中心線の作成

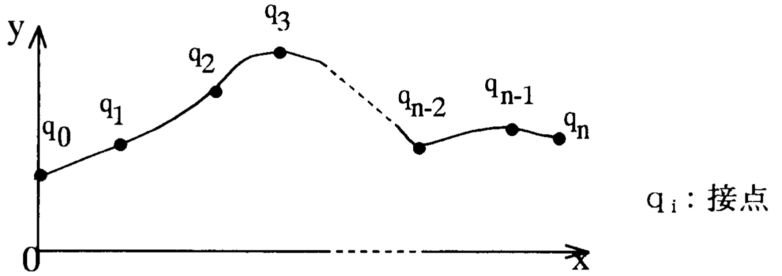


図4　スプライン関数

また $p_i(q_i)$ は

$$p_i(q_i)=(q_i-q_j)(q_i-q_{j+1})\ldots(q_i-q_{i-1})(q_i-q_{i+1})\ldots(q_i-q_{j+K})$$

と表すこととすると，ドブァ・コックスの漸化式

$$B_{i,K}(x)=\frac{x-q_i}{q_{i+K-1}-q_i}B_{i,K-1}(x)+\frac{q_{i+K}-x}{q_{i+K}-q_{i+1}}B_{i+1,K-1}(x) \tag{5}$$

を求めることができる。

そして，スプライン関数は

$$S(x)=\alpha_0B_{0,K}(x)+\alpha_1B_{1,K}(x)+\ldots+\alpha_{N-1}B_{N-1,K}(x)=\sum_{i=0}^{N-1}\alpha_iB_{i,K}(x) \tag{6}$$

と表すことができる。

3.3 パラメトリックスプライン関数

文字を $x-y$ 平面に表したとき，ある x に対して y がいくつか存在する場合がある。したがって，このような場合には一価関数の式では表すことが出来ない。そこで，任意の座標 (x,y) に対して一意に決まるパラメータを導入することによって多価関数を実現する方法がある。その一つとしてパラメトリックスプライン関数がある。この関数はパラメータ ξ を用いて

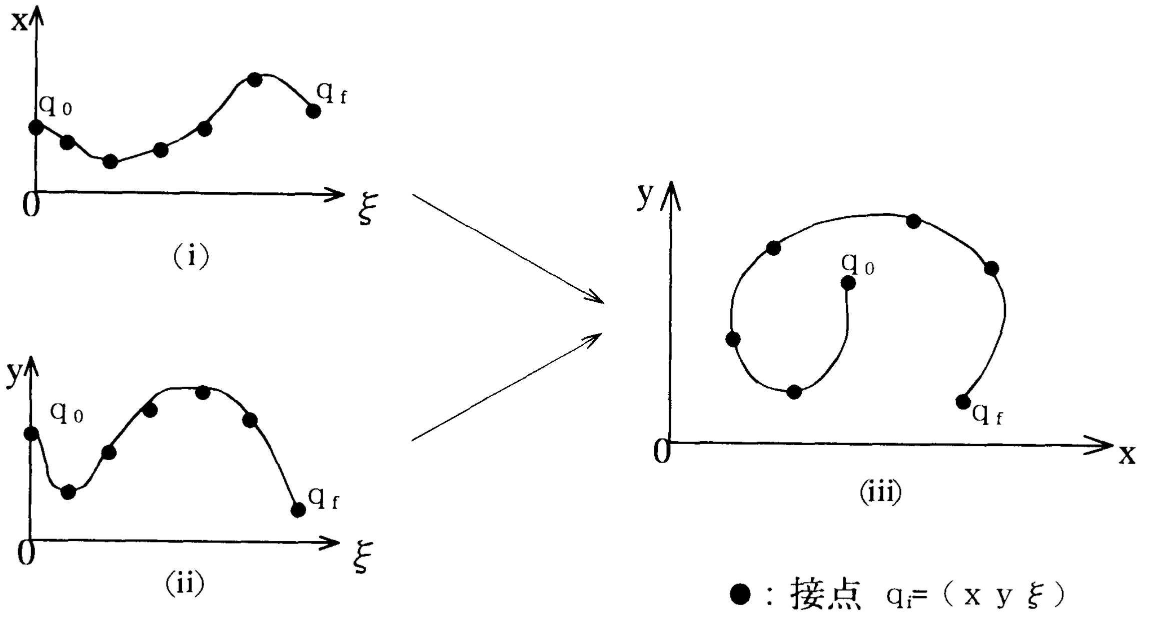


図5 パラメトリック スプライン関数

$$\begin{cases} x = x(\xi) \\ y = y(\xi) \end{cases} \tag{7}$$

と表すことができる。$x-y$ 平面上の接点は座標 (x, y) におけるパラメータの値 ξ を用いて表されるので，パラメトリックスプライン関数の構成は(6),(7)式を使用すれば容易である。

スプライン関数の次数は制御系設計問題において，加速度情報が必要な場合が考えられる。その連続性を考慮して[4]，ここでは3次のスプライン関数を構成している。

これを(6)，(7)式を使って表すと，

$$\begin{cases} x(\xi) = \sum_{i=0}^{N-1} \alpha_i B_{i,4}(\xi) \\ y(\xi) = \sum_{i=0}^{N-1} \beta_i B_{i,4}(\xi) \end{cases} \tag{8}$$

となる。ここで(8)式の α_i, β_i は未知数であるが，それ以外 (x, y, ξ, q) は既知であるから，(8)式を α_i, β_i についての連立方程式として解くことができる。

こうして求めた α_i, β_i を使って補間を実行すると例えば，図5(i)(ii)のように表すことができる。そしてそれを合わせると図5(iii)のように $x-y$ 平面上で多価な軌道を実現させることができる。

4．軌道生成結果

ここでは先に述べた方法で軌道生成した結果と，文献1）の最小二乗法による軌道生成結果を比較し，提案する軌道生成法の有効性を示す。また，軌道生成可能なひらがなの数が増加するということを示す。

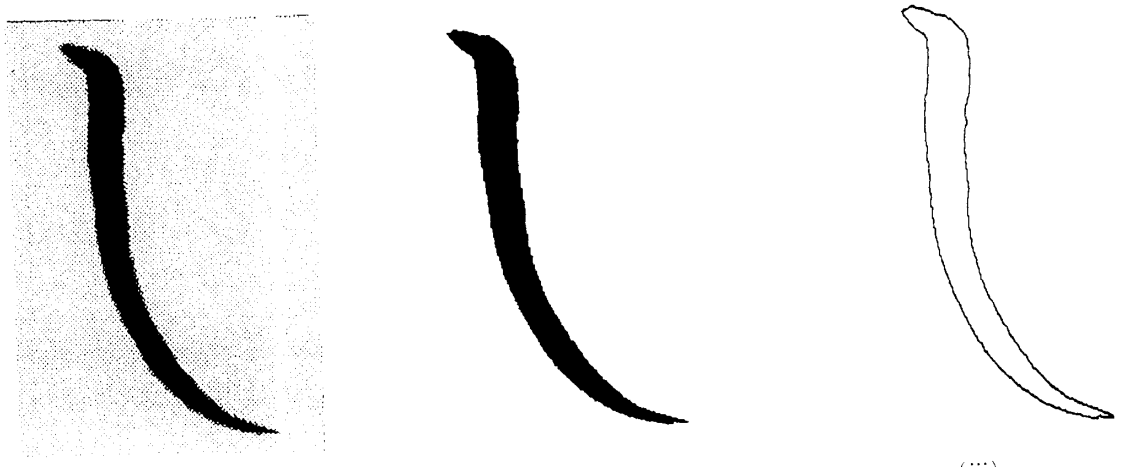
4．1 最小二乗法との比較

提案する軌道生成法（パラメトリック スプライン関数）と最小二乗法の2つの方法で軌道生成の結果をひらがな「し」を例に取り以下に示す。なお，パラメトリック スプライン関数の最高次数は3次，最小二乗法の最高次数は8次である。

図6はひらがな文字「し」の(i)取り込み画像を，(ii)2値化，(iii)エッジ検出と画像処理した結果である。

図7に中心線データと任意に○印を5点，△印を12点選んだデータを示す。

図8よりパラメトリック スプライン関数の方が中心線データに近い軌道生成が行われている。そして，最小二乗法が大量のデータと次数を必要とするのに対してパラメトリック スプライン関数では非常に少ないデータ数と3次の次数でかなりよい軌道を実現することが図8より確認できる。



（ｉ）　（ii）　（iii）

図６　画像処理結果

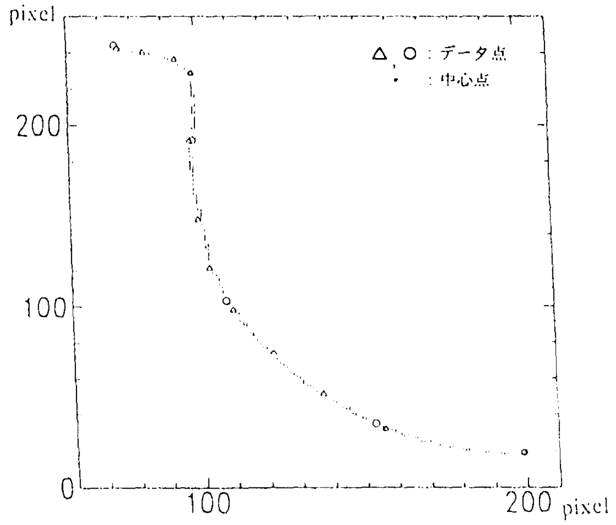


図７　中心線データ

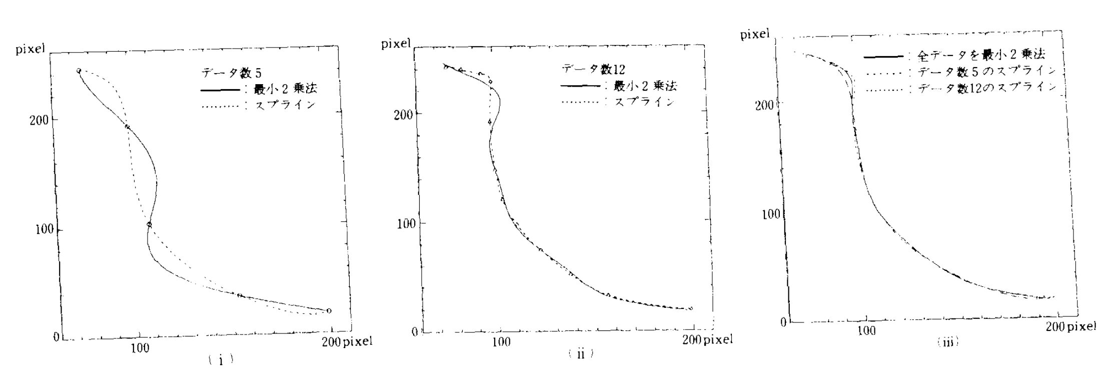


図８　パラメトリック スプライン関数と最少二乗法による軌道結果

4.2　多価な座標を持つひらがなの軌道生成

次に，多価な座標を持つひらがなの軌道生成の結果を数例示す。

軌道生成の条件として

1）始点と終点が軌道上の他の点と交差していない。

2）一筆書きである。

3）交差点を持たないひらがなである。

以下の文字は，一価の多項式間では軌道生成は不可能であるが，スプライン関数を使っての補間ではかなりよい軌道生成の結果を得ることができている。

また，一価の多項式補間では一筆書きひらがな13文字中 4 文字（く，し，つ，へ）が軌道生成可能であるのに対し，スプライン関数を使った場合ではさらに 6 文字(そ，て，の，ひ，ろ，ん）の軌道生成が可能である。

5．おわりに

中心線の抽出方法とパラメトリック スプライン関数を使って一筆書きひらがなの軌道生成法を述べてきた。この方法を使うことによって手本の中心線データ座標中の一部のデータを使ったにもかかわらず，

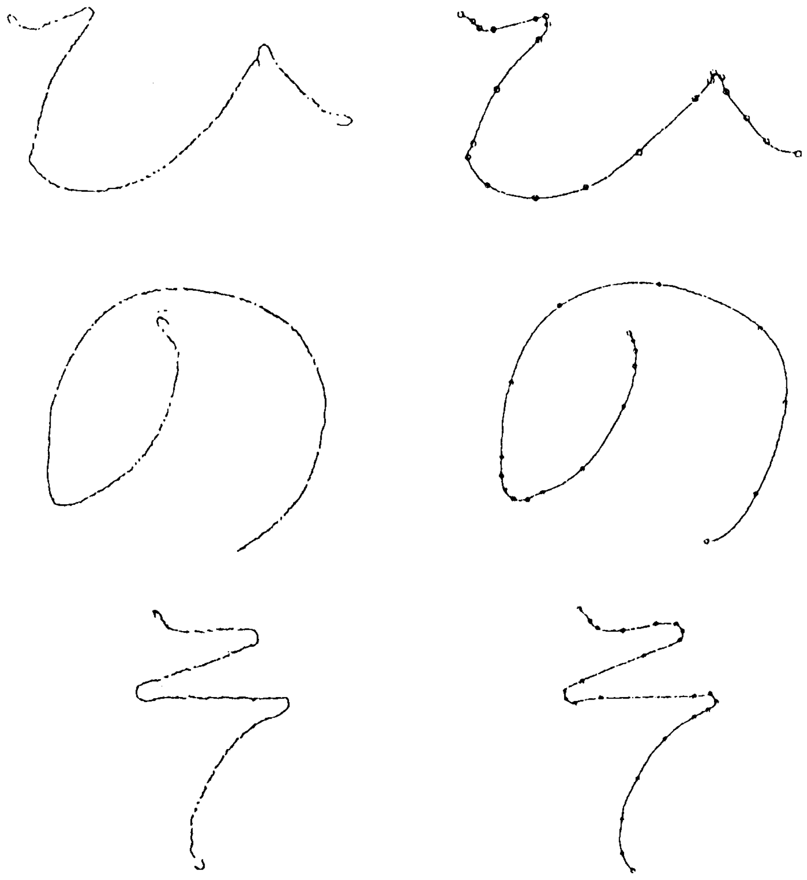


図 9　軌道生成結果

1．生成される軌道は最小二乗法を用いた軌道生成より低次で誤差の少ないよい結果を得ることができた。

2．軌道生成可能なひらがなの数を増加することができた。

という結果が得られた。

参考文献

1）西口，藤本，逢坂：書道ロボット開発に関する一実験，日本機械学会，中国四国支部第34期総会・講演会講演論文集，No. 965-1，255-256（1996）.

2）長谷川，輿水，中山，横井：画像処理の基本技法<技法入門編>，技術評論社，34-37，41-44(1986).

3）桜井，菅野，吉村，高山：Cによるスプライン関数，東京電機大学出版局，15-36，68-89（1993）.

4）川崎：ロボット工学の基礎，森北出版株式会社，134-137（1991）.

付　　録

細線化の操作を軌道の中心線データを求めるために用いた例を図10に示す。

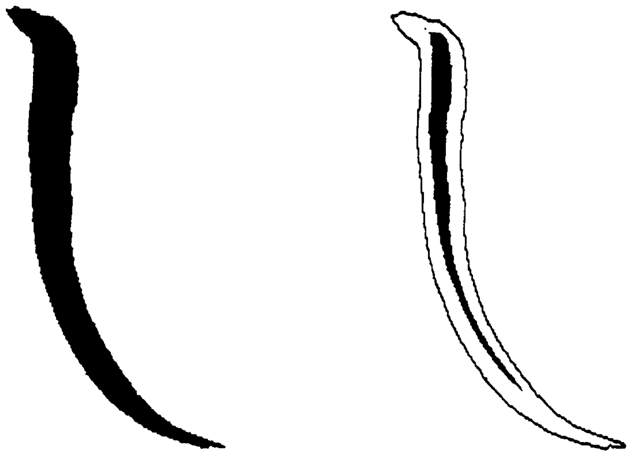


図10　細線化

Study of Calligraphy Robot

—— Creation of Hiragana Letters ——

Hiroya OKE*, Shinsaku FUJIMOTO** and Kazumasa OHSAKA**

**Graduate School,*

***Deparment of Mechanical Engineering,*

Faculty of Egineering,

Okayama University of Science,

Ridai-cho 1-1, Okayama 700 Japan

(Received October 7, 1996)

We can write letters with brush simply. We look the reference letters in our eyes or we find the reference letters from our memory. We recognize the letter in our brain and find a center line and create the path. Next we transmit the imformation in the way of our arm and we actully write a letter. We execute these processes.

If these processes are executed by the robot, the robot takes in a data of the refrence letters with a camera. The robot executes image processings by computer. The robot recognizes a reading data from the memory unit. And the robot creates path. The robot transmits the imformation in the manipulator's controller. The robot writes a letter by controlling the manipulator.

There are important things in the processes, problem of letter recognizing, problem of path planning, problem of controller planning.

Now we deal with the problem of path planning. We consider a center line is a important factor for letter's shapes. We have executed the path planning for "hiragana" by the minimum square method to realize a center line. However we can get only "hiragana" pathes that they can be expressed by one-to-one correspondence coordinate by the minimum square method. And we have a problem in that data quantity of the path and dimension of the approximation expression of the curve are very large. Further we have other problem in that all the pathes don't through the center line, so they through off the path of reference letters. Therefore we propose the path planning method that uses data of center line and parametric spline function. We try to decrease error between the center line of reference letters and that obtained by our proposed method, and increase the numbe of "hiragana" possible path planning.