

書道ロボットに関する研究

—平仮名文字の書き順生成と軌道計画—

藤本 真作・奥田 誠二*・逢坂 一正・小野 敏郎

岡山理科大学工学部知能機械工学科

*株式会社牧野フライス製作所

(2005年9月28日受付、2005年11月7日受理)

1. 緒言

近年、ロボットの性能が向上し、正確で精密な作業が可能になってきたことから幅広い分野で利用されるようになった。しかしながら、現在のロボットには人の持つ細かい感覚や高度な精密さを必要とする作業は困難であり、産業界では繰り返し作業などの単純な動作を行っているのが現状である。

人が実現しているこれらの作業は、作業者がその作業を繰り返し行うことで「経験」や「こつ」などの言葉もしくは、数学的に表現することが困難な情報の蓄積によって実現できるようになったと考えられる。このように、人が作業を熟達する過程で得る知的な情報を、工学ではスキル(skill:技能)といい、これを定量化もしくは定式化し、ロボットシステムに導入することが可能であれば、ロボットにも人と同等の作業が実現できるものと考えられる。

本研究では、スキルをロボットシステムに導入するにあたって、スキルが必要な作業として「書道」という作業に注目し、筆文字を書くロボットシステムを構築することを目的としている。本目的を実現するためには、まず手本から文字を認識し、ロボットを動作させる(文字を書く)までの過程をオペレータの介入なしに自動化させる必要があると考えられる。そこで、本研究では特に文字認識部すなわち文字の書き順生成と軌道計画に重点をおき、そのアルゴリズムの構築を目指す。また、書道のスキルのひとつである運筆速度に着目し、ロボットアームを用いて文字を書く場合の運筆速度の速度パターンを生成し、その適用方法を検討する。本手法では、カメラから取り込まれた文字画像からその構成要素や速度パターンなどによって文字の軌道を生成するため、文字書体・書き順・大きさなどの膨大なデータベースを作成・保存する必要がなく、参照データ量の低減や、様々な文字に対して適用することができる。

書道における作業を分類すると大きく分けて以下の3つの過程に分類することができる。まず、頭の中で文字の軌跡を考え、つぎに、文字の形状などから速度などの書くために必要な情報を選定し、最後に実際に腕を動かして文字を書く。これらの過程を工学的なアプローチに置き換えると、文字認識問題(書き順・軌跡生成問題)、軌道生成問題、制御系設計・実装問題に置き換えることができる。本論文ではロボットの自動化に最も重要と考えられる文字認識問題および軌道生成問題について議論する。

2. 書き順生成問題

手書き文字の軌跡を作成するためには、与えられた文字の書き順を考える必要がある。図2.1に示すように、文字は、「交差点・端点」と「線」の部分からなる図形としてみることができる。グラフ理論において、このような図形のことを「グラフ」¹⁾という。「交差点・端点」のことを「頂点」、「線」のことを「辺」、そして、頂点につながる辺の数を「分岐数」という。このグラフ理論には、オイラーの定理²⁾すなわち「グラフが一筆書きできる条件は辺が奇数本接続している点がかたかだか2つである」というものがある。言い換えると、分岐数が1である始点と終点以外の頂点が偶数分岐数であれば、始点から終点まで一筆書き可能であるといえる。書き順生成問題ではこの定理を利用し、認識すべき文字の各ストロークに関してオイラーの定理を適用できる形にすることが、大半の目的となる。認識する文字は平仮名文字(71文字)を対象とし、この平仮名文字の書き順に起こりうる問題を考える。すべての平仮名文字に対してオイラーの定理が適用できる形にするためには、以下の3つの問題点が挙げられる。

【問題点】

1. 「の」や「べ」のように始点や終点が辺などと重なっている場合、始点や終点が認識できない。
2. 「ん」などのように一筆書きできない文字（2重線）が存在する。
3. ストロークの数が複数存在する。

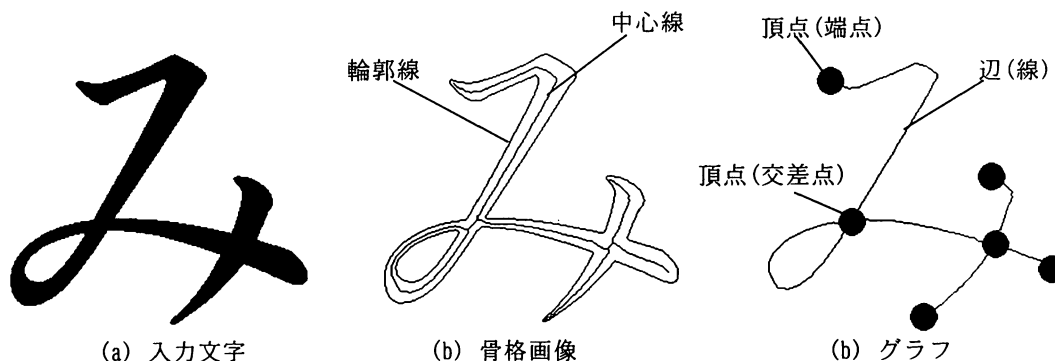


図2.1 入力文字画像のグラフ化

2-1 始点・終点の認識

各ストロークを一筆書き可能なグラフにするためには、始点と終点以外の頂点が偶数分岐数であり、分岐数1の始点と終点が必要となる。しかし、平仮名文字には始点や終点が認識できない文字が存在する。このような文字は以下に示す2つのケースに分類することができる。

- ・ 始点あるいは終点が辺に重なっている場合で、図2.2(a)のように端点が辺に重なっている部分に注目すると、図2.2(b)のような直交点に近い形になっていることがわかる。そこでこの角度の大きさ θ_1 、 θ_2 、 θ_3 の組み合わせがある範囲内であれば最大角度の対辺を分離し、分離した頂点を始点もしくは終点として認識することにする。ただし設定したシミュレーション条件下において、始点または終点の誤認識をなくすため、本論文では角度の範囲をそれぞれ $\theta_1 \in [52, 91]$ 、 $\theta_2 \in [86, 120]$ 、 $\theta_3 \in [157, 203]$ とした。
- ・ 始点と終点が重なっており、閉じた線になっている場合である（図2.3(a)）。この場合は、閉じた線になっているため、この辺につながる頂点が存在しない。このため、この辺につながる頂点を作成する必要がある。そこで、閉じた辺の一番上の部分に頂点を作成し（図2.3(b)）、その頂点を複製・分離する（図2.3(c)）ことで、これらの頂点を始点と終点として認識する。

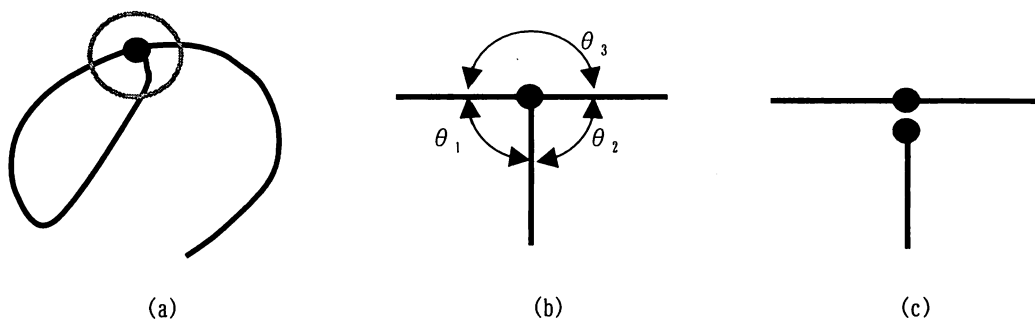


図2.2 端点が辺に重なっている場合

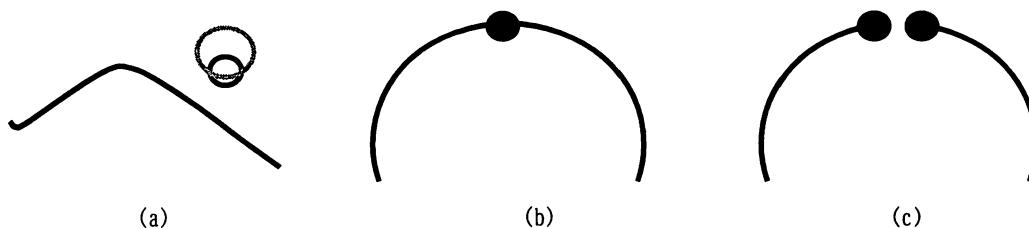


図2.3 始点と終点が重なっている場合

2-2 2重線の認識

オイラーの定理を適用することができないため、一筆書きできない文字すなわち2重線の存在が挙げられる。このような文字を一筆書きできる形にするためには、オイラーの定理より、1ストロークの始点から終点まで、始点・終点を除く頂点を偶数分岐数にする必要がある。そこで、図2.4のように、奇数分岐頂点に挟まれた辺を二重線(図2.4(a))すなわち2本の辺(図2.4(b))として考え、2度通るようになることで解決できる¹⁾²⁾。ただし、自然な流れで書かれる傾向として、奇数分岐頂点において最小角度 θ の対辺が2重線になるようにする必要がある。このように、2重線を考えることでそれにつながる奇数分岐頂点を偶数分岐頂点にし、オイラーの定理が適用できる形にすることができる。

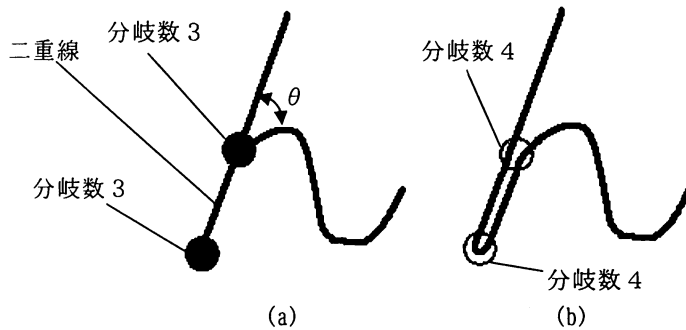


図2.4 一筆書きできない文字

2-3 辺トレース

前小節において、オイラーの定理が適用できる形になったと考えられる。そこで、ここでは文字のストロークすなわち画の認識をする。まず、ストロークの始点を決め、分岐数1の頂点まで辺を辿っていき、その頂点を終点とする。そして、まだ辿ってない辺が存在すれば、まだ辿っていない頂点を新しいストロークの始点として決め、分岐数1の頂点まで辺を辿り、その頂点を終点とする。このように、辿るべき辺がなくなるまで繰り返す。これで、グラフを構成するすべての辺を辿り、各ストロークの認識に代える。

ここで、辺を辿る途中に3以上の分岐数を有する頂点が発生する場合がある。このとき、この頂点において、入ってきた辺からつぎにどの辺を選んで辿るかが問題となる。ここでは、滑らかさ・自然な流れで書かれる傾向から、入力辺と出力辺で角度変化が最小になる辺を辿ることにする(図2.5)。図2.5の例では角度が最小となる順路3の辺を辿ることになる。

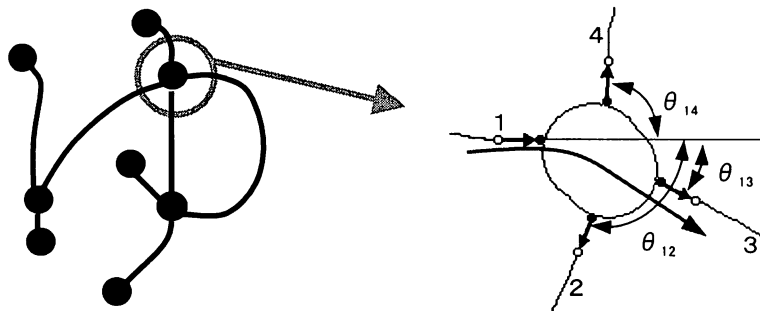


図2.5 辿るべき辺の選択方法

2-4 書き順決定

2-3節において文字の各ストロークが認識できたため、つぎにこれらのストロークの順番すなわち書き順を決める必要がある。この書き順の決定には各ストロークの重心を利用する。まず、各ストロークの重心位置を求める。そして画像に対して指定された一定な傾き角の直線を左上から右下へ移動させ、この直線が各ストロークの重心位置と重なった順番をストロークの順番(書き順)とする。ただし、一定の傾き角は書き順生成の正解率が最も高い65.5 deg.の値を採用している。また、濁点などのように似た形であり近い距離のストロークはひとまとめに考えることにする。濁点をひとまとめに捉えることによってほぼ自然な書き順にすることができる。ストローク重心位置によって書き順を決定した結果を図2.6に示す。図中の数字と位置はその文字の書き順と始点の位置を表している。記号 \odot はストロークの重心位置である。

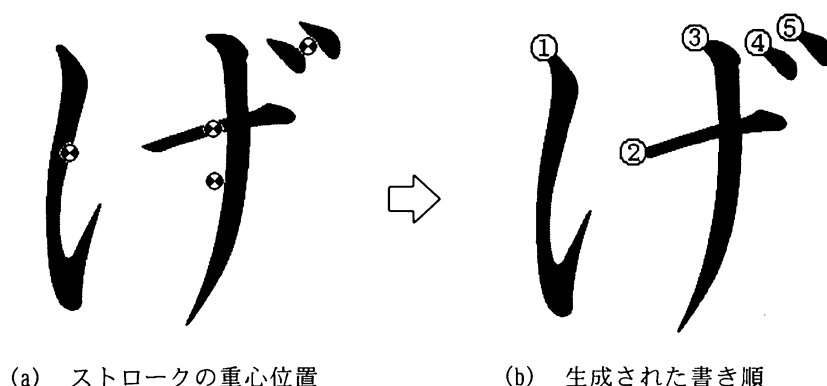


図2.6 重心位置による書き順の決定

3. 軌跡の生成

マニピュレータの手先目標軌跡を生成するためには、書き順情報だけでなく文字の書き始めから書き終わりまでの連続的な軌跡が必要である。しかし、ストロークとストロークの間すなわち筆が宙に浮いている状態での筆先の軌跡は文字画像から直接的には分からない。そこで、まずストロークとストロークの間の補間を考える。そして、これらのデータから文字の書き始めから書き終わりまでの軌跡の生成を行う。目標軌跡の生成にはパラメトリックスプライン関数によって低次元で滑らかな目標軌跡の補間を行い、作業面（紙面）に設定した $x-y$ 座標の軌跡と、前もって測定した筆の沈み量と文字の太さの関係式から紙面に垂直に設定された z 方向の軌跡が得られる。求められた軌跡を組み合わせることによって筆先の3次元軌跡を生成する。

3-1 ストローク補間

前章で得られた書き順情報から、目標軌跡を作るためにはストロークとストロークの間である、筆が紙面から離れているときの軌跡情報が必要になる。しかしながら、手本文字画像からこの軌跡を得ることはできない。そこで、人が実際に文字を書いている状態をカメラによる3次元位置を計測し、運筆傾向を見出すことにする。図3.1に計測結果の一部を示す。図3.1(a)に図の詳細を示す。図の左側には計測位置の $x-y$ 平面座標を、図の右側には図3.1(a)に表記したように z 方向の高さ情報を表している。この高さは筆が浮いているときの筆先端の位置データで、横軸は終筆からつぎのストロークの始筆までの直線距離であり、縦軸は高さである。うえから順番に筆順に沿って現れる筆脈の順序になっている。

計測の結果を見ると、筆の浮いた部分の特徴として以下のような特徴が見られる。

- ・ $x-y$ 平面に関しては、ほぼ直線によって最短距離の移動をしている。
- ・ z 方向に関しては上に凸の放物線を描いている。
- ・ 最大高さは、終筆部から始筆部までの直線距離に比例している。

以上のことより、複数のストロークでできている文字に関して、ストロークの間は $x-y$ 平面に関しては直線で、 z 方向の高さに関しては上に凸の2次曲線により軌跡を作成することにする。

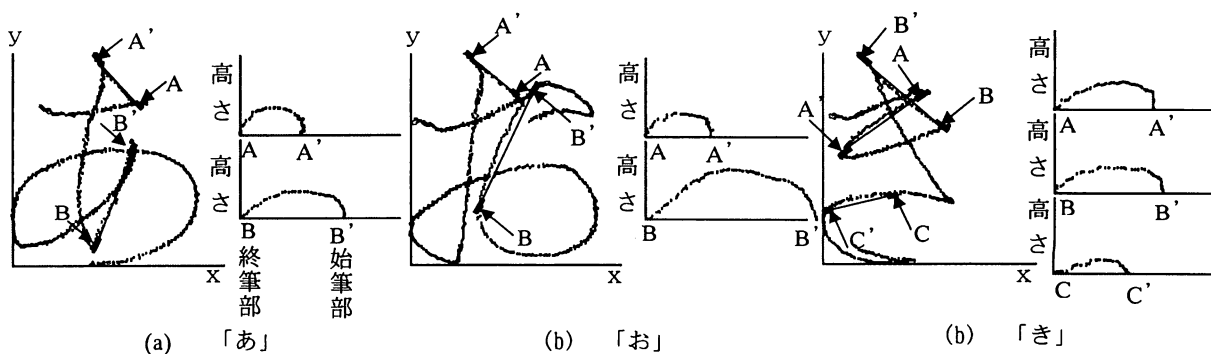


図3.1 3次元位置計測結果

3-2 パラメトリックスプライン関数による軌跡補間

作成された書き順情報と文字の軌跡を滑らかで低次元な関数で表すことができるパラメトリックスプライン関数によって目標軌跡の補間を行う。

4. 目標軌道の生成

3章で生成した書き順情報を含む目標軌跡を利用して、マニピュレータを実際に動かすための目標軌道を生成する。目標軌道を生成するためには、作業空間における目標軌跡に加え時間を考慮する必要がある。時間情報の導入には、書道のスキルのひとつである筆の緩急（速度パターン）について考慮する。

4-1 要素分割・速度パターン

文字は直線や曲線といったある曲率半径を有する構成要素に分割することができ、この曲率半径によって速度パターンは変わると考える³⁾⁴⁾。ここで速度パターンとは、文字の運筆速度および経過時間を最大運筆速度と最終時間でそれぞれ割り、無時限化された運筆速度の形状をいう。そこで、各曲率半径による速度パターンデータ³⁾（図4.1）を利用し、文字の速度パターンを作成する。速度パターンは書道の有段者、数名から抽出したデータによって作成されている⁵⁾。また、速度パターンおよび、時間に関しては汎用性や利便性の観点から無次元化されている。はね・はらい・空中部分に関しては、人が文字を書くときの傾向から、ほかの要素よりも速度を速くすることにする。

要素の分割の例として図4.2に文字「の」を取り上げる。ここで分割された各部位を速度パターンと曲率半径の関係を用いてつぎのような名称で定義する部位に分ける。

○：判定点（文字を構成する要素ごとに分割するために条件をあてはめる点）

●：変曲点（文字を構成する要素ごとに分割する点）

Aに相当する部位：曲率半径の異なる構成要素

構成要素A₁からA₄の曲率半径に対応する速度パターンを図4.1のデータから抽出する。文字「の」の場合では4つの構成要素に分割され、表4.1のように4つの速度パターンが抽出されることになる。

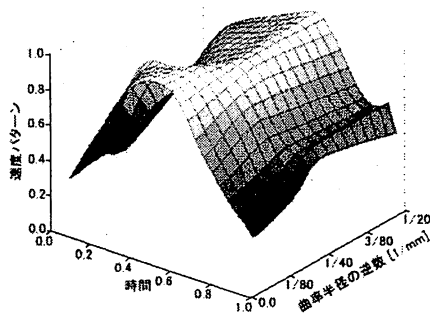


図4.1 速度パターンと曲率半径の関係

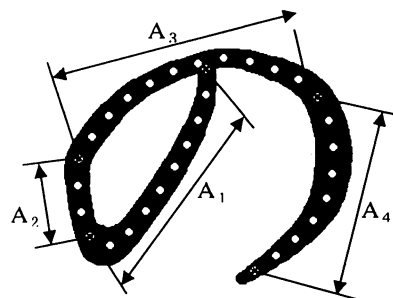


図4.2 文字「の」の要素分割

表4.1 文字「の」における各要素に対応する速度パターン

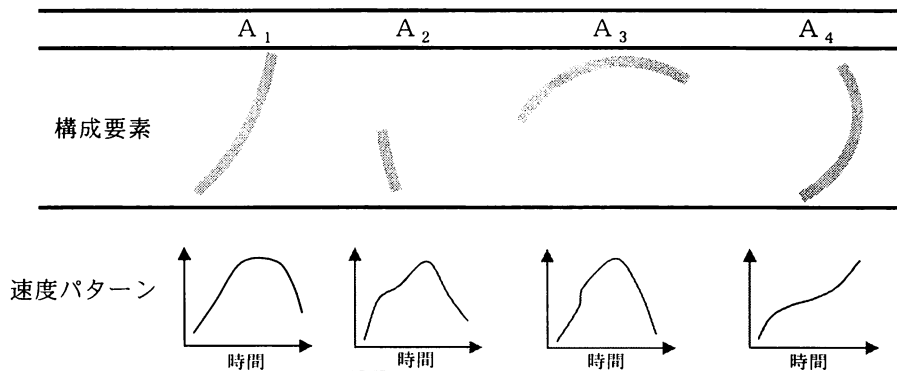


図4.1から抽出される速度パターンには時間情報が含まれていない。そこで本論文では、既知である構成要素の長さと最大運筆速度の関係式から最大運筆速度を推定する⁹⁾。この手法により構成要素の運筆速度が算出でき、個々の運筆速度を繋ぎ合わせることで図4.3に示すような文字全体の運筆速度を生成することができる。

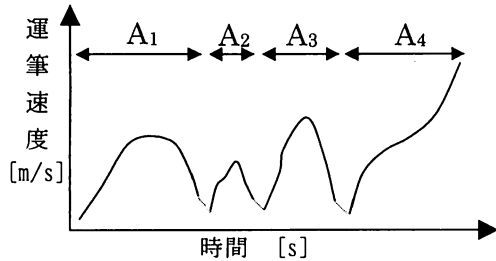


図4.3 文字「の」全体の運筆速度

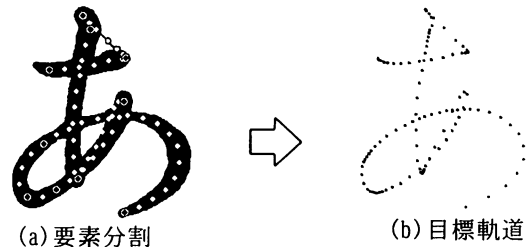


図4.4 目標軌道の生成

4-2 目標軌道の生成

軌道追従制御を行うためにはロボットアームに与える目標軌道を生成する必要がある。そこで、サンプリング周期ごとに生成した運筆速度からその速度の値を読み取り、運筆速度の積分値と目標軌跡の長さが等しくなった点の座標によって目標軌道を生成する。得られた目標軌道から逆運動学問題を解くことで各関節角度が導出される。導出された関節角度を目標値として軌道追従制御を行うことでロボットに文字を書かせることができる。図4.4(a)は文字「あ」を要素分割した様子を表し、黒抜きの点が構成要素と構成要素との変曲点を表している。一定のサンプリング毎に軌道をプロットしたすなわち時間経過を考慮した目標軌道が図4.4(b)であり、要素の繋ぎ目部分が密に（運筆速度が遅く）、払い部分が疎に（運筆速度が速く）なっていることが確認できる。

5. 書道における平仮名文字生成実験

書道における文字生成実験は、平仮名文字「あ」から「ん」までの濁点を含める71文字を1文字ずつ実行させる。本来、カメラから文字画像を取り込み処理するが、ここでは再現性・一般性の問題から文字画像はパソコンで生成することにした。フォントはHG正楷書体-PRO、色はグレースケールを使用した。

あらかじめ用意された手本画像のうちのひとつを選び（図5.1(a)）、文字画像は書き順を生成するために細線化され、骨格画像を生成する（図5.1(b)）。その後、グラフとして認識し（図5.1(c)）、提案したアルゴリズムによって書き順の生成を行う（図5.1(d)）。そして、書き順生成結果が一般に決められている書き順と一致した場合のみ軌道生成を行う。書き順に沿った軌跡データから曲率半径による構成要素に分割する（図5.2(a)）。そしてその要素に対応する速度パターンを速度パターンと半径の関係グラフから抽出し、それを繋ぎ合わせることで文字全体の運筆速度を生成する（図5.2(b)）。指定されたサンプリング周期で目標軌道を生成し（図5.2(c)）、実際のロボットマニピュレータによって平仮名文字を生成する。

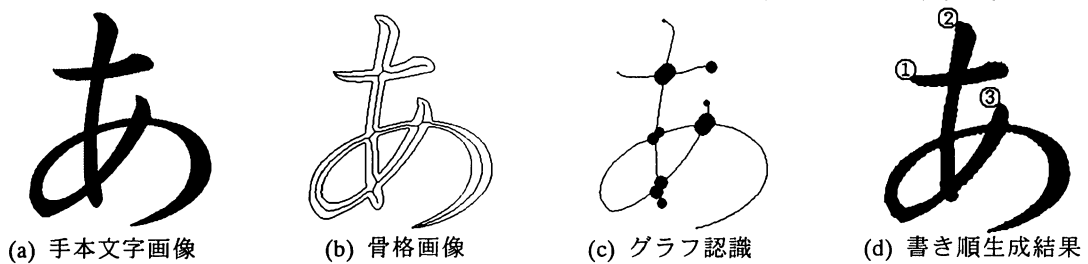


図5.1 書き順生成過程

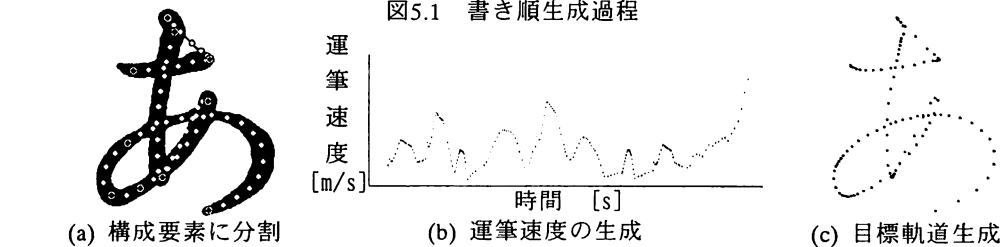


図5.2 目標軌道生成過程

5-1 文字の書き順生成結果

書き順生成結果に関してはすべての文字の書き順は生成することができた。しかし、71文字中12文字、「か」「が」「せ」「ぜ」「た」「だ」「な」「ふ」「ぶ」「も」「や」に関しては習字において決められている書き順とは異なる書き順となった。濁点・半濁点を除くと全平仮名文字の1割に相当する7文字が書き順とは異なる結果となった。例として図5.3に本方法によって得られた書き順の一部を示す。この書き順の違いは、「平仮名」を知らない人が文字を見た場合、2-4節で述べた重心位置の左上からの順番に近い書き順を想像する可能性があると考えられる。しかしながら、これらの文字の書き順生成に関しては、再度アルゴリズムの検討が必要である。

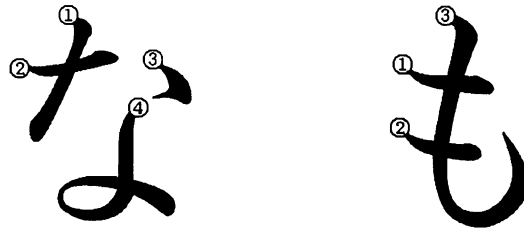


図5.3 本方法による書き順生成結果

5-2 運筆速度の比較

生成された運筆速度の妥当性を検証するため、実際に人が書いた運筆速度との比較を行った。図5.4に運筆速度の比較結果の一例を示す。ここでは、2画以上で比較的複雑な形状をした文字を選定した。図5.4は左から要素分割された文字、生成された運筆速度、計測された運筆速度の3種類を示している。

生成した運筆速度と、実際に人が書いた運筆速度を比べると、ほぼ同じ形状・パターンになっていることが確認できる。この形状・パターンはスキルの一種と考えられるため、書く人が異なったり、人が同じでも状況が違ったりすることで多少の差異が生じる。このことから、極めて類似した運筆速度が得られていることが確認でき、本手法の有効性が確認できた。

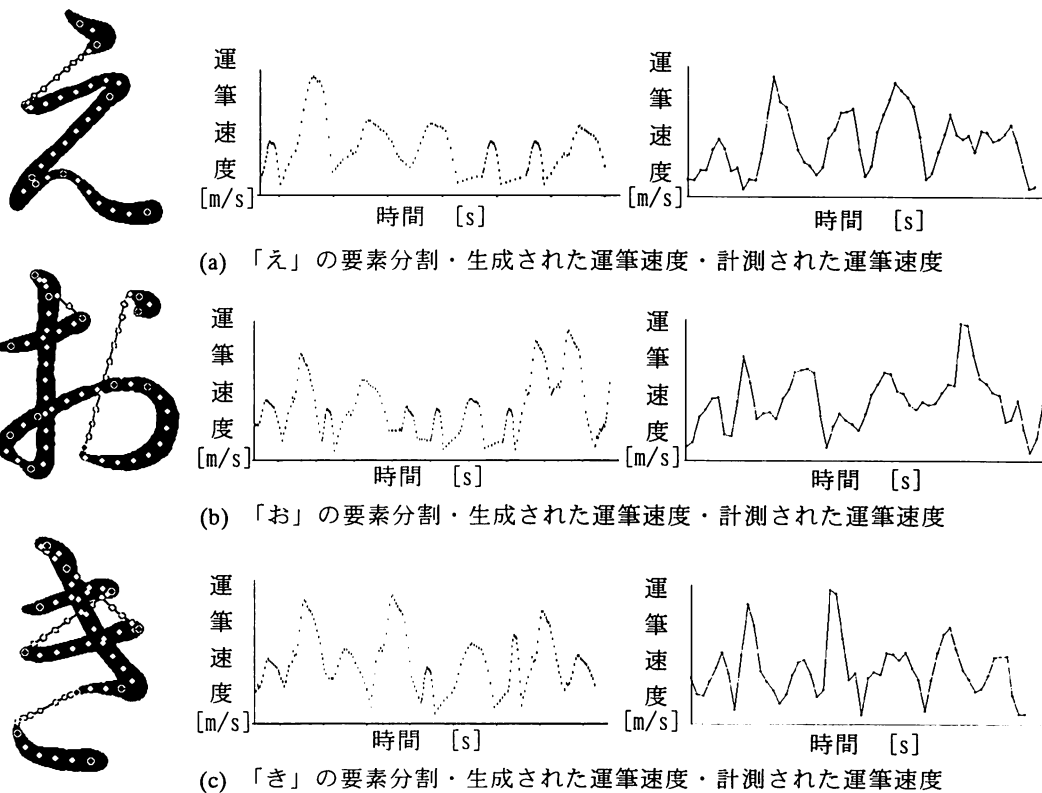


図5.4 運筆速度の比較結果

5-3 マニピュレータによる文字生成の実験結果

三菱重工製7自由度マニピュレータPA10を用いて書道運筆時における速度パターンの適用実験を行った。サンプリング周期は20msecで、エンドエフェクタに取り付けられた筆が目標軌道を追従することで、作業平面に文字が書かれる。文字を書くときのマニピュレータの手先姿勢は、筆が紙面に対して常に垂直になるように扱う「直筆法」とする。図5.5(a)に手本とした文字、図5.5(b)にマニピュレータによって書かれた文字の一例を示す。生成された文字はお手本に近い文字を実現していることが確認でき、書道におけるスキルの一部としてみなせるものが解明・適用できたと考えられる。しかしながら、図5.5(b)より生成された文字には入筆部や止めの要素に問題点が残されていることが分かる。これらに対する新たなスキルの導入は今後の課題である。

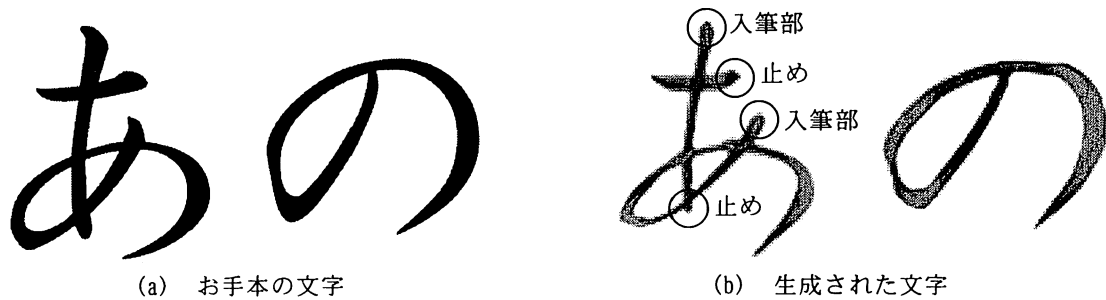


図5.5 マニピュレータによる文字生成

6. 結言

文字をグラフとして置き換えることで、一筆書き問題の拡張として、平仮名文字画像から文字の書き順を生成し、書道における文字生成の自動化を実現した。対象文字は複数ストロークも考慮されているため、始点から終点まで可能な書き順を列挙し、すべての組み合わせを評価する場合、膨大な量の組み合わせすなわち、組み合わせ爆発が生じる可能性がある。そのため提案するアルゴリズムは書き順の列挙は行わず、簡単な手法によって文字の書き順生成を実現することができた。また、書道におけるスキルのひとつである人が文字を書くときの運筆速度に近い形状・パターンの生成も実現した。最後に、これらの情報を利用し、マニピュレータを用いて文字を書き、人の動作に近い緩急をつけた文字を書かせることができた。以上のことから、人のもつスキルの一部としてみなせるものが解明・適用できたと考えられる。

今後の課題として、習字における書き順とは異なる平仮名文字（12文字）に対する新たなアルゴリズムの検討と、入筆部や止めの要素に対するスキルの導入などが挙げられる。

参考文献

- 1) 佐藤公男：“グラフ理論入門—C言語によるプログラムと応用問題—”，日刊工業新聞社，pp.2-11, pp.28-37 (1999)
- 2) Tong Huang, Makoto Yasuhara：Recovery of Information on the Drawing Order of Single-Stroke Cursive Handwritten Characters from their 2D Images, Trans. of IPSJ, Vol.36, No.9, pp.2132-2143(1995.9).
- 3) 片岡，新家：岡山理科大学工学部卒業論文 書道ロボットに関する研究—毛筆文字の軌道計画について—(2001)
- 4) 山本：岡山理科大学大学院工学研究科修士論文 書道ロボットに関する研究—書道運筆時の速度パターンの生成・解析とその応用—(2002)
- 5) 山崎：岡山理科大学大学院工学研究科修士論文 書道ロボットに関する研究—書道運筆時における速度パターンの抽出とその応用—(2000)
- 6) 奥田：岡山理科大学大学院工学研究科修士論文 書道ロボットに関する研究—平仮名文字の書き順生成と軌道計画—(2003)

A Study on Calligraphy Robot

—The stroke order of the Japanese cursive syllabary (Hiragana)
and the trajectory planning—

Shinsaku FUJIMOTO, Seiji OKUDA*, Kazumasa OHSAKA
and Toshiro ONO

*Department of Intelligent Mechanical Engineering,
Faculty of Engineering,
Okayama University of Science*

1-1 Ridai-cho, Okayama 700-0005, Japan

**Makino Milling Machine CO.,LTD.*

Mimase 359-3, Aikawa-cho, Aikou-gun, Kanagawa 243-0308, Japan

(Received September 28, 2005; accepted November 7, 2005)

This paper describes a method to generate the stroke order of a Japanese cursive syllabary from a static 2D image and to apply a velocity pattern of brush in writing letters such as human skill to calligraphy robotic system. Humans can write beautiful letters with a brush easily. Many practice and experience are necessary for conducting this. However it is difficult that robots can write beautiful letters, because written letters need human skill. Skill is necessary to execute advanced human-like task for robots.

In this paper, we selected calligraphy from the advanced tasks with skill. In our opinion, skill has the method of character recognition and the velocity pattern of the brush in writing letters. Therefore, the stroke order of character is considered by dividing the character into elements and velocity patterns. This method does not need to use various database, because use of a few velocity patterns data and the stroke order generated by algorithm, and it is possible to apply this method to various characters. The Japanese cursive syllabary written by human is treated in this method.

After the character is scanned in and preprocessed, we generate a graph of character that is composed of vertex and edges by skeletal image, and the graph is traced from the start vertex to the end vertex using the incidence matrix information at each stroke. The character is divided into elements from radius curves, and velocity patterns are determined by relations between velocity patterns and radiuses at each element. The velocities of elements are applied to velocity patterns and connect each velocity pattern.

This paper is organized as follows: In chapter 1 introduces the background of this paper and human skill. In chapter 2, preparations are presented to generate the stroke order. In chapter 3, the drawing order generation problem is solved. The drawing order path is generated by a function approximation. In chapter 4, the trajectory planning is given by means of a velocity pattern. Chapter 5 covers an application of the skill to robotic system and the experimental results. In chapter 6, the conclusions of the proposed method are discussed.