

ペン入力文字の連綿毛筆体を含む高品質書体への変換技術の高度化

A system to change handwriting characters into the traditional cursive style

大場 幹浩
Mikihiro OHBA

〒441-8008 愛知県豊橋市三ツ相町401 E-mail: o@o.email.ne.jp

ABSTRACT. In the system to change oriental handwriting characters into the traditional cursive style, the simulation quality was extremely improved. The basis of the simulation to change to classical calligrapher's styles was created, and for practical use of the generated image in the pen computing, the sub-system to relate handwriting characters with the text code was added.

1. 背景

個人の年賀状から企業広告に至るまで、日常の様々な局面で多様な書体が必要とされているが、現状では一般に、ユーザはメーカーから供されるパッケージフォントからその好みや用途に適する製品を選択するしかなく、使用可能な書体の種類は限られたものになる傾向があった。

さらに、かな草書体などの、文字間を接続するラインを含む書形の表現は、フォントを用いる方法では一般に実現が困難であり、デジタル機器上での柔軟で加工可能な形でそれは他の何らかの方法によって達成されなくてはならないと考えられる。

また近年、ペン入力を標準とするタブレット型コンピュータが複数の企業から市販されているが、そのハード的機能を生かすソフトの少なさが課題となっている。

2. 目的

開発者は従来から、ディスプレイ上にマウス等で描いた任意の手書き文字を、その文字軸を構成する座標点に特定の線形変換を作用させることによって一定の品質を持ったその崩し系書体に変形させるソフトウェアの開発を行って来た。そのシミュレーションの質を向上させることにより、前段の課題、すなわち、容易かつ多様に形成され得る、自筆の文字を元にした書体群の一般用途での活用、連綿書体の自動生成、及びペンコンピューティングそのものの活用性の向上に供する成果を得ることが出来るものと考えた。

3. 概要

3-1. 書体変形の基礎方法

文字変換の基礎的な方法について最初に述べる。

1. 行書系化・草書系化などの形状の基本的な「崩

し」変換に関しては、独自開発の中点連鎖法アルゴリズムを基盤としたライン（画、紐）の変形と接続に関する一連の方法論（superstroke theory）によって、それら伝統的な書体に近接的な滑らかな曲線群に、元の軌跡の座標列そのものを変換する。

2. 文字の「太さ」や「掠れ」、「連綿（隣接線の見た目のつながり）」などのシミュレーションに関しては、1で変換を行った軌跡の座標列への、筆先形などのペイントパターンの付加の仕方の調節によってそれを行う。

3. 最後に、生成された文字ラインの輪郭線に対して、オリジナル・既成の双方の方法論を含む精細なスムージング処理等を行い、文字画像の最終的な品質を上げる。

3-2. 従来例

3-1の技術を各々素朴に適用することによって実現していた、従来の自作ソフトによる毛筆体シミュレーションの例を図2の中央図に示す。

崩れの状態を含め、一般的な崩し文字に近似的な様態は持っているが、形状・線質ともに工夫を図るべき余地の多いものでもあった。

3-3. 開発目標

本提案の作業目標として

- 1) 変換書体の質的向上
 - 2) 古典的書家の書体のシミュレーション
 - 3) 手書き文字とテキスト文字との対応登録、主要機能のライブラリ化などのソフトウェア的整備
- の3点を設定した。

4. 成果

4-1. 変換書体の質的向上

研究にあたっては、3-2の例のような従来のシミュ

レーション文字と一般の毛筆文字とのグラフィカルな相違点について検討を行い、質の向上には主に以下の点がその要になると考え、それを実現する変換アルゴリズムの構築と実装を行った。

1) ライン輪郭線のキレの向上

毛筆ラインの線質レベルのシミュレーションは、3 - 1 - 2 に記載のように、対象となる文字の軸ラインに対して筆型のパターンでそれをなぞっていくことによって行っているが、それを単純に実装する方法では、図1の左図のように一般に輪郭線にある程度の凹凸が付くことになる。

この輪郭線のスムーズ化にも、軸線の変形時と同系の手法を一種のスムーズ化アルゴリズムとして使用しているが、その適用法に工夫を加えることにより、結果として図1の右図のようなキレのあるラインがほぼ自動的に得られるようになった。



図1：線質に鋭角性を付加するオートスムージング

2) 元ラインの様態を考慮した崩し強度の調整

従来の素朴な崩し変形によるものと、一般の毛筆崩し文字との形状面における相違について検討を行い、

- ・ラインの向きによる崩しの強さの相違
- ・元ラインの長さによる崩しの強さの相違
- ・屈曲の複雑さによる崩しの強さの相違
- ・転折部の形状の明瞭化（崩しの低減化）

等の点に注目し、文字内の部分的な構造の相違を認識するとともに、その崩し強度を調整する機構の付加を行った。

3) 元ラインの様態を考慮した太さ値付けの調整

文字ライン各部の太さ変化の様相について検討を行い、

- ・ラインの長短
- ・転折部
- ・単線性

等の、部分的構造を考慮した太さ値付けの調整用アルゴリズムを構築し、その実装を行った。

4) 装飾的处理

文字形状の補足的・装飾的な処理について検討を行い、

- ・隣接線の配置関係による接続性の変更
- ・文字全体の右上方向への傾斜化
- ・ライン先端部への微細な導入線（ヒゲ）の付加

等を表現する機構の実装を行った。

これら1)～4)の新しい機構を用いた変換シミュレーションの例を下図に示す。図2～5とも、最左図がオリジナルの手書き文字であり、右図が変換例である。



図2：元文字 基礎的崩し変換 新機能を用いた変換



図3： 元文字 変換例



図4： 元文字 変換例



図5：元文字 連綿体への変換 掠れの付加

これらは各々の元文字に対するごく標準的な変換の一例であり、プログラム内では崩れの強度や太さ等をはじめとする数多くのパラメータを各々多段的に調整可能である。

4 - 2 . 書家風書体へのシミュレーション

基本的な崩しの強度や太さの出力値、さらには上記4 - 1 内の様々な項目を、著名な書家の書の特徴に近接的なものになるようにシステム設定したコマンドをプロトタイプソフトウェア内に実装した。

現状では書家の数も各々のシミュレーションの質の調整もまだ十分とはいえず、ごく原型的なものであるが、ユーザによる任意の手書き文字を、様々な書家風の書体に1クリックでシミュレートする機構の基盤になるものである。

また、さらなる便宜のために、一つの文字列に対する同シミュレーションの結果を、複数の書家分をまとめてhtml形式で一括表示出来るコマンドもプロトタイプソフトウェア内に装備した。

4 - 3 . ソフトウェア的補助機構

毛筆書体生成に関するソフトウェアとしての利便性を向上させるものとして、以下のような機構の実装を行った。

1) 手書き文字のファイル登録

画面上に描いた手書き文字に対応するテキスト文字をキーボードから入力し、両者を対にしてディスク上に記録する、つまり、手書き文字データの辞書ファイル的な登録を行うことにより、一度描いた手書き文字の出力を、以後はテキスト入力によって行える機構をプロトタイプ内に実装した。

なお、ソフトをタブレットPC上で動作させる場合には、同OSに付属の文字認識ライブラリを利用して手書き文字をそれに対応する文字コードに自動対応させる機構の実装も行ったため、同OS上での使用に際しては、テキスト文字のユーザによる指定を基本的に省くことが可能である。

2) 主要機能のライブラリ化

毛筆体生成機能の他ソフトからの使用を可能とするための主要ルーチンのライブラリ化を行った。

現在、Microsoft社のVisual C++コンパイラで制作されたソフトであれば、そのプログラム内からライブラリコール用のオブジェクト(下例では syo)に対して、

- 1 . syo.create() 関数でライブラリを初期化
- 2 . syo.set_points() 関数で、マウスなどの軌跡として取得した元文字の座標点列を設定
- 3 . syo.set_parameter() 関数で「崩し」「太さ」などの変換用パラメータを設定
- 4 . syo.get_image() 関数で上記設定に基づいた毛筆体の画像をライブラリに生成させ、それを取得

と4つの関数のみを使った操作を行うことにより、ごく一般的に毛筆生成アルゴリズムを扱うことが出来るもの

とした。

3) その他の補助機能

生成された毛筆文字画像のSVGベクトルファイルへのコンバートを可能とした。これにより生成画像のXMLフォーマットファイル内での扱いが容易になった。

5 . 要約および今後の課題、展望

1) 文字品質向上の基盤となる技術の研究と実装

基本となるこの目標に関しては、多くの考察とそれを実現するためのコマンド群の実装を行い、旧来のバージョンに比べて生成される文字品質の一定の向上化を行うことが出来たと考えている。

ただし、表現細部の調整機構等が未だ完全なものであるとは言えず、太さのバランスに欠ける箇所が発生する等、その改善を行うべき点を持つ上に、またそもそも、「良質な「書」のシミュレーション」という課題自体、そのレベルアップの作業に簡単な終わりがあるというタイプのものでは本質的にないため、今後も引き続きその質的向上化の作業に尽力していきたいと考えている。

2) 書家風書体へのシミュレーション

その基本機構の実装を行ったが、書家の数もその質的調整も十分ではなく、今後その改善及び拡張を行いたいと考えている。

3) ソフトウェア的補助機構

前段で言及の通り、その基本機構の実装を行った。

4) 毛筆フォント等のビジュアル書体へのシミュレーション性の向上、および楷書体へのシミュレーション

なお、市販の行書体フォントなどの、ビジュアルライズされた書体に近接的な毛筆体を手書き文字から自動生成することを遠望的な目標の一つとしていたが、それそのものを1コマンドで生成するレベルの機構の制作には至らなかった。しかし、4 - 1で言及のコマンド群によってキレのある線質や形状を一般的に持たせ得る方向になったため、生成される文字一般が、ある程度まではそれらに近似的な書体と見なし得ることが多くなったのではないかと考えている。

また一方、文字ラインの全般的な直線化や、全体のバランスを考慮した部分構造の再配置を行うことにより、その楷書体へのシミュレーションを行う企図も持っていたが、期間内での実装には至らなかった。

双方ともに今後その改善及び実装化作業に尽力していきたいと考えている。

