草書体漢字の計算機生成と認識支援行列

竹内　良亘

いわき明星大学　理工学部　電子工学科

あらまし 漢字の行書体から草書体を生成する方法を示す．行書体漢字を一筆書きで表現した線図形を多数の均等切片に分割し，それぞれの切片に筆が動いて行く先の数十個の切片をベクトル的に足し合わせると草書体漢字が生成される．足し合わせる切片の数は行書体の画数に依存し，全体を256分割した場合，画数が多い字で約15個，少ない字で約100個である．この切片を足し合わせる操作を行列で表現したものを生成行列と呼ぶ．その逆行列を草書体漢字にかけると行書体が復元される．この場合，生成行列により生成された草書体の切片は均等長さではなく，それぞれ異なる固有の長さを持つものになっている．任意に与えられた均等長さ切片の入力草書体に対しては，逆問題の近似解法により原行書体を推定しうる程度に復元できる．本報告で提案する草書体漢字の生成行列は，2つのパラメータで広範囲な草書体漢字の生成と原字推定に有効である．

Computer generation of cursive Chinese characters and its recognition by a matrix

Yoshinobu Takeuchi

Department of Electronics Engineering
College of Science and Engineering
Iwaki Meisei University

abstract A computer-aided procedure to generate cursive Chinese characters is investigated with matrix method. The superposed image of line segments of semi-cursive Chinese characters manifests cursive style. The addition of segments is performed using forward segments in the range of 15 to 100 in the case of total 256 segments. The matrix representation of the procedure is one of characteristic measure of cursive Chinese characters. A possible mean for overcoming the difficulties of recognition of cursive style is illustrated using the inverse matrix.

1. まえがき

草書体漢字は，古典原本の研究者か書家でないと解読は困難である．まして計算機に認識させるには，草書体専用の計算機辞書がない限り絶望的である．この辞書の作成は，上記専門家の研究創作領域のみならず，筆跡による個人識別の計算機技術に不可欠のものであり，今後必要度が高まると予想される．しかし，一つの漢字に対する草書体の変形度の多様さは，草書体漢字の表現モデルの決定版がいまだない現状と合わせて，辞書作成の困難さを示唆している．

草書体から行書体の復元という困難な問題に取り組むには，その前にまず草書体漢字の生成の問題を解決しなければならない．それは生成過程の逆過程が行書体を復元させるから，生成過程の検討が不可欠であるからである．草書体漢字は，筆の動きが先を急ぐあまり省略体として描かれたものであるから，筆の動きが重要なのは論を待たない．ここでは既に紙の上に描かれた字体を扱うこととするので，草書体漢字の生成モデルを作る際の基本的な考え方は，筆の動きをいかに筆跡の静的モデルに取り組むかである．

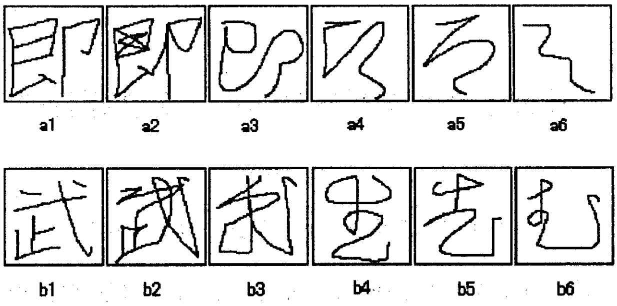

図 1: 草書体漢字の多様性の例

以下に 2. おいて草書体漢字の特徴を述べた後，3. において生成モデルとしての行列法を示す．次に 4. において生成例を示す．最後に 5. において逆問題を解くことにより草書体漢字から行書体を復元できる可能性を述べる．

2. 草書体漢字の特徴

まず始めに行書体漢字とその草書体の例を図 1に示す．図 1の a1 が「即」の行書体，b1 が「武」の行書体であり，活字体である楷書体をほとんど崩さずに書いたものである．a2 および b2 は，a1, b1 を書くとき本来空中を通るペン先の軌跡も紙面に記した一筆書きである．a3 から a6 ならびに b3 から b6 は，この一筆書きから派生した草書体である．これから見て分かるように草書体は変形の程度が小さいものから大きいものまで多様である．容易に行書体を類推できるものから，全く別記号と見えるものまで存在する．注意しなければならないのは草書体の中には，行書体の省略筆法では説明できない字体があることである．篆書体から派生した字体がそのような例である．そのほか楷書体ないし隷書体の略体を崩してできた草書体もあり，そのような略体が用いられた時代に通用した草書体である．

このような草書体が入力文字として与えられた場合，同一カテゴリーの原字体として一つの行書体を設定できないことになる．極端な場合，平仮名を草書体漢字としてみたとき，一つの平仮名に対する原字体は何種類かある．また草書体は似ているが原字体が異なるものも多い (図 2に示す a3 と a33，b3 と bb3 の例)．このように草書体漢字と行書体漢字は一対一の対応ではないことは念頭に置いておくべきである．

3. 草書体漢字の生成モデル

3.1 曲線記述子

草書体漢字が生成される過程は，行書体の速書き，省略筆法からできる崩し字体が基本となっている．多くは一筆書きになっており，筆点が紙面を離れず描かれる．従って草書体漢字の表現モデルとしては一筆書き曲線モデルが適している．それに対応して行書体も一筆書き曲線で表現して原字体とする．図 1の a2,b2，図 2の a2,aa2,b2,bb2 の例に示すように，行書体漢字を書くときに筆点が空中を動いて描く軌跡も実線として表現する．このように行書体と草書体を共に一筆書き曲線で表現しておけば，両者の対応は一本曲線の変形あるいは変換で扱えることになる．

一筆書き曲線を計算機で扱えるように表現するには，始点から終点までの各点の座標を列に並べればよい．しかし，この表現は各点が固定されているので，平行移動した曲線を同じ曲線として扱えない．この依存性を無くするために隣り合う点同士の差をとって表現する．$(z_1 - z_0, z_2 -$

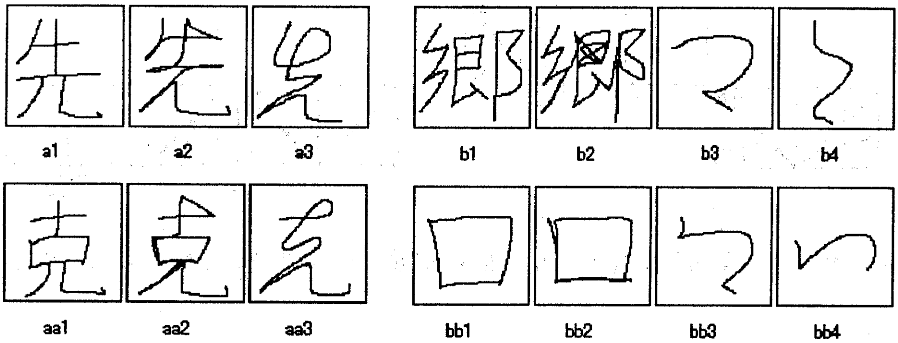

図 2: 似た草書体漢字と対応する行書体漢字

$z_1, \ldots, z_n - z_{n-1}$) ここで z_k $(k = 0, 1, \cdots, n)$ は複素数で表した k点の座標である．ここで $w_k = z_k - z_{k-1}$, $(k = 1, 2, \cdots, n)$ と書くと一筆書き曲線は $(w_1, w_2, \ldots, w_n)$ と書き表される．

この表現は，曲線を n 個の切片に分割し，それぞれの切片を複素ベクトル w_kで表したことになっている．切り取る点は曲線が方向を変える点にとる．したがって w_kの長さは一律に同じではない．分割数 n を十分大きくとれば，例えば 256 分割にすると均等分割でも十分な精度で表現できる．

3.2 生成モデル

一筆書きで与えられた行書体漢字から草書体を生成するにはどうするかを考える．ここで扱う草書体は，行書体の速書きにより省略された送筆の結果できるものに限定する．変化の激しい曲線を緩やかな近似曲線に変換するには，切片列 $(w_1, w_2, \cdots, w_n)$ を区分的な区間で平均するという手段が考えられる．しかし，この方法では，先を急いで出来る曲線を表現するという特性が反映されていない．速書きするときの意識としては，筆点が現在ある点から次ぎに書かれるであろう先々の行書体の曲線が現時点に取り込まれている．従って，行書体という変化の激しい曲線から草書体という柔らかい曲線へ変換するには，行書体の筆跡の n 個の切片に先々で書かれるであろう m 個の切片を取り込めばよいと考えられる．(m は自分自身も含めた個数である．) すなわち次のような変換をすればよい．

$$w_1 \leftarrow w_1 + w_2 + \ldots + w_{1+(m-1)}, \quad w_2 \leftarrow w_2 + w_3 + \ldots + w_{2+(m-1)}$$
$$\cdots \quad w_{n-1} \leftarrow w_{n-1} + w_n, \quad w_n \leftarrow w_n \tag{1}$$

最後から $m-2$ 番目までは，先々の切片の個数が減っていくので，加える切片数も減っていき，最後の n 番目の切片は変換前と等しい．行列で書くと

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}_2 = \begin{bmatrix} 1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 1 & 1 & 0 & \cdots & 0 \\ \vdots & & & & & & & \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & 1 & 1 \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}_1 \tag{2}$$

この方法により各切片が変化していくという捕らえ方が出来る．上式右辺の行列を生成行列と呼ぶ．

まず最も単純な場合を示す．原字が 2 本の曲線からなる場合の例を図 3a に示す．元の曲線の

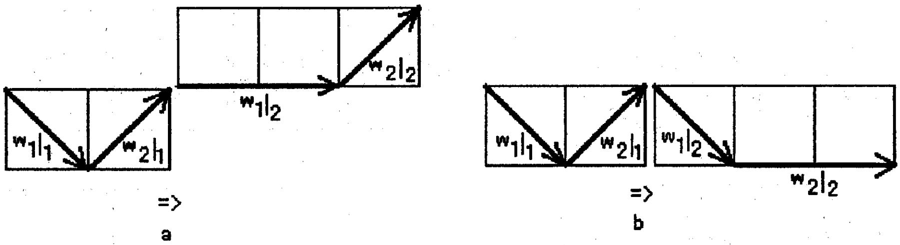

図 3: 切片数 $n = 2$, 取込み数 $1(m = 2)$ の場合の変換. a:先の切片の取り込み. b:後の切片の取り込み.

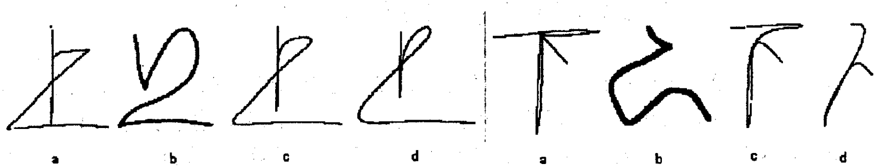

図 4: [左側]a:「上」の行書体の一筆書き入力, b:辞典の草書体の一筆書き, c:生成草書体 $(m = 20)$, d:生成草書体 $(m = 40)$. [右側]「下」について同様に a:, b:, c:$(m = 40)$, d:$(m = 100)$.

出発位置を変換後も同じとすると, 変換された曲線は上方へシフトする. ここで比較のために, もし, 取り込む切片を過去の直前の筆跡から1個選ぶとすると図3b となる.

図3a と図3b から分かるように, 取り込む切片を先々で書かれるであろう切片にすると原字の一筆書きの終点近傍が強調され, 取り込む切片を過去の切片から選ぶと原字一筆書きの始点近傍が強調される. 草書体漢字は図3a に示す終端近傍が強調された場合にあたり, 先々の切片から取り込むべきであることが分かる.

式(2) で示される基本生成モデルは, その行列構成を変形することにより, 種々のタイプを作ることが出来る.

[1] 取り込む先の切片を飛び石的に取り込む.

[2] 取り込む切片に一律の重み gを掛ける.

これらの変形モデルは, 基本生成モデルでは対応できない場合に解決法を与える.

4. 草書体漢字の生成例

3.2 節で述べた基本生成モデルにより, 入力行書体漢字から草書体を生成した例を以下に示す. 入力行書体としては, 丁寧に一筆書きで作成した手書き漢字とし, その一本曲線を均等に256 等分した切片の両端座標を入力データとした. 生成した草書体漢字が妥当なものであるかどうかは, 書体の辞典 (伏見冲敬, 書道辞典, 角川書店 (1977)) に記載されている歴史的な墨跡を一筆書きの線書きに改めたものと比較して判断した. 図4から図7に示す例では, a が入力行書体, b が辞典の草書体の一筆書き, c,d が本方法で生成した草書体2種類である. これらの生成草書体は, 生成モデルの式(1)(2) における m を変えて得られたものである.

図4では, 辞典の字体は生成書体の幅を広げたものになっている. 生成書体は速書きからできる草書体を表現できている. 図5では, a の入力行書体の筆順を「口」の部分で変更している. 速書きに対応して右回りに「口」を書いたものとして入力した. この結果図5の左側では辞典の字体

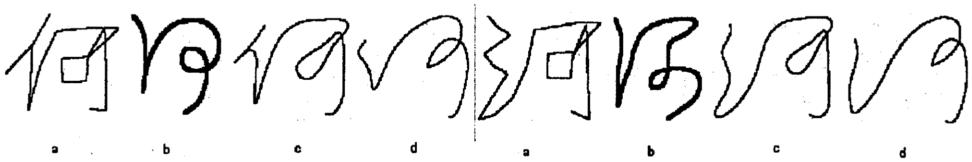

図 5: [左側]a:「何」の行書体の一筆書き入力，b:辞典の草書体の一筆書き，c:生成草書体 ($m = 20$)，d:生成草書体 ($m = 40$)．　[右側]「河」について同様に a:，b:，c:($m = 20$)，d:($m = 40$)．

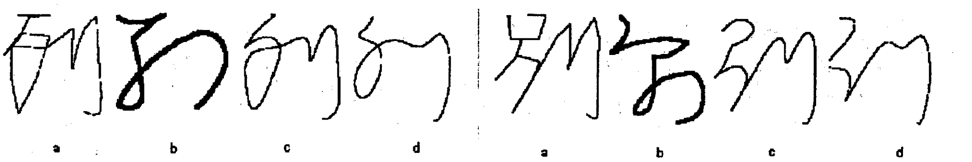

図 6: [左側]a:「列」の行書体の一筆書き入力，b:辞典の草書体の一筆書き，c:生成草書体 ($m = 20$)，d:生成草書体 ($m = 40$)．　[右側]「別」について同様に a:，b:，c:($m = 20$)，d:($m = 35$)．

がほぼ生成されている．図 5の右側では,「可」の部分が辞典書体と生成書体とで異なっている．辞典書体では図 5の右側と左側とで「可」の部分の崩し方が異なっている．本生成法では全て同じに図 5左側の崩し方になる．図 6左側では，草書体の最初の突っ込み部分を除いて辞典の字体が生成されているといえる．図 6右側では辞典書体の生成はできていない．しかし，速書きとしての崩し字体は生成できている．図 7では，辞典草書体とは同じではないが，全体としての特徴がよく似ている字体が生成されている．

以上の例から言えることは，歴史的な墨跡として残されている草書体漢字にも変形度が大きいことを考慮すると，本方法で生成された草書体漢字は，標準的な草書体の字体を生成しているということが出来る．草書体漢字を生成できるモデルが出来たことは，手書き崩し文字を扱う第一ステップを確実にしたことになる．

5. 草書体漢字から行書体の推定

3. で述べた生成モデルの特徴は，入力一筆書き曲線を単に滑らかに補間したのではなく，筆点の運動を反映すると共に入力切片の情報を捨て去らずに保持していることである．これは草書体漢字が与えられたときに，もとの行書体を推定できる可能性を示唆する．

それは式 (2) において $[w]_2$が与えられたとして $[w]_1$を求めることである．この場合次の量が不明であることに注意しなければならない．

(a) 与えられた草書体を生成したであろう式 (2) の行列が不明である．すなわちサイズ n と取り込む範囲 m が不明である．

(b) 式 (2) で生成された $[w]_2$の各切片のそれぞれの固有の長さが不明である．すなわち図 3からも分かるように，$[w]_1$の切片の長さが均等のとき，生成された $[w]_2$の各切片の長さは均等長さでなくなるが，これはあらかじめ与えられない．

従って次のような手順で式 (2) の逆変換を試みることになる．

① n と m を適当に仮定する．

② 入力草書体の一筆書き曲線を均等に n 分割する．

③ 式 (2) の行列のかわりに，変形モデル [2] を用いてパラメータ gの調整作用を利用する．

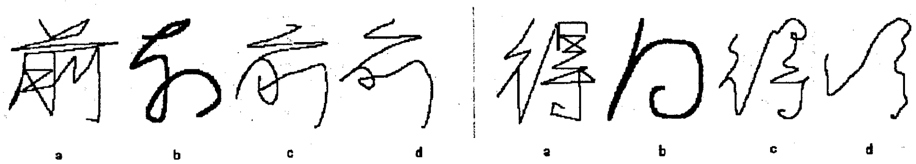

図 7: [左側]a:「前」の行書体の一筆書き入力，b:辞典の草書体の一筆書き，c:生成草書体 ($m = 20$)，d:生成草書体 ($m = 35$)． [右側]「得」について同様に a:，b:，c:($m = 15$)，d:($m = 50$)．

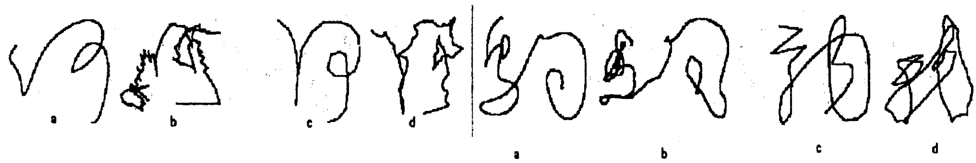

図 8: 入力草書体の原字体の推定例．[左側]a:図 5 の d で与えられる「何」の入力草書体．b:推定される原字体．($n = 256$, $m = 50$, $g = 6$)．c:辞典の草書体．d:推定される原字体．($n = 256$, $m = 20$, $g = 0.2$)． [右側]「駒」について同様に a:($n = 256$, $m = 15$, $g = 1$)，b:($n = 256$, $m = 20$, $g = 0.2$)，c:，d:($n = 256$, $m = 30$, $g = 0.06$)．

④ 足される入力切片数は行列の下の部分で順に一つずつ少なくなる事を考慮する．
⑤ 式(2) において与えられる $[w]_2$は，上の対応付けに従うと，傾きの情報を与えるのみで均等の切片長さは意味がないことになる．そこで求める逆変換曲線 $[w]_1$の各切片の長さが均等になるように求める．行列方程式の下の行から順に解いていく．

図8はこのようにして入力草書体からもとの行書体を推定した結果である．図8の [左側]b では，左辺の「にんべん」が再現されておらず，また「さんずい」と推定するのも困難である．右辺の「可」は復元されている．d では入力字体を多少角ばらせた程度にとどまっている．図8の [右側]b では，原字が何とか推定できる程度に復元されている．d は復元できていない．このように与えられた草書体漢字から行書体への復元が困難な場合が多いのは，上述した (a) と (b) が未知のためである．上の例では n を仮定して 2 つのパラメータ m, gの 2 次元平面での最適値探索を行っている．正解はあらかじめ分からないのであるから，ダイナミックプログラミングの手法は使えない．復元の困難さの最も大きな原因は，(b) に述べたように，3. の生成モデルでは切片長がふぞろいになることである．

6. むすび

一筆書き曲線で与えた行書体漢字から，先々の筆の動きを取り込むことにより草書体漢字が生成できることを示した．この方法は行列手法として表現できるため，草書体漢字から逆変換で行書体を推定する方法ともなることを示した．その場合，入力草書体から原字体を推定するために，生成行列のいくつかのパラメータを変動させて最適値を見いだす必要がある．各種パラメータ値による逆変換の結果を並べたものが草書体漢字を認識する支援法になっている．本方法は筆跡による個人識別法へも応用が可能であり，今後の展開が待たれる．