

ストローク数非依存の高速オンライン手書き文字認識手法

4D-4

レ・バン・トゥー 秋山 勝彦 中川 正樹  
(東京農工大学)

1. はじめに

本稿ではストロークのつながりを許すパターンマッチング手法と、漢字の偏や旁を有する構造化字体表現を組み合わせ、オンライン手書き文字を高速かつ、高認識率で認識できる方式を提示する。このマッチング手法とサブパターン辞書及びその学習機能で、オンライン手書きの続け字、書き順変動の文字等に適切に対応でき、日本語の広い範囲の文字を高速に認識できると期待している。

2. 設計方針

本方式では漢字字体を構造化して、この構成単位で様々な筆順を表現しておいて、認識時には文字ごとにすべての可能な筆順パターンを生成して、マッチングする。

漢字を構造化する段階では、漢字を偏、旁等（全てを部首と呼ぶ）に分割する。ある文字・部首を分割するか、しないかの厳密な基準はないが次のような幾つかの判断により行う。

・分割できそうな文字と部首を分ける。

例：仕 = (人偏) + 士, 維 = 糸 + (ふるとり)

・二つの部首からなると見られるが、ユーザが、書く時に第一部首をまだ書き終わらないうちに第二部首の幾つかの画を書いてしまう可能性が多い場合は分割しない。

例：国 > < 口 + 玉, 果 > < 田 + 木

・二つ部首のうち一つは他の文字と共有されない、かつ複雑な部首だと見られ、もう一つが書き順変動があまりないようだと見られるならば、分割しない。

例：燕, 叡, 劓, 慧

分割するには、漢字の第一水準の全体（2965文字）を中心として行う。結果は次の通りである  
漢字第一水準：2965個, 中間部首：593部  
分割した文字：2557個, 分割した中間部首：328部  
分割しない中間部首：265部, 分割しない文字：408個  
文字と中間部首の全体に関して

全体の文字と部首： 3558部

分割した文字と部首： 2885部

分割しない文字と部首： 673部

分割しない、文字・部首を基本部首と呼ぶ。

辞書では基本の文字と部首に違う書き順のパターンを登録する。分割した文字・部首は特別の場合だけパターンがある。字体表現を構造化しない辞書と違って、本辞書では頻繁に多くのパターンを一時的に作る。基本部首から再帰的二分木構造によりパターンを作る。基本部首に典型的な書き順のパターンを入れるならば、様々な書き順のパターンに対応できると考えられる。

3. 基本構造とパターン演算子

本節では漢字の再帰的二分木パターン結合とパターン演算子について述べる。上に述べたように基本部首にはパターンが幾つかあるが、多くの漢字・部首は基本部首に分けられる。

ある漢字 A に対して、パターンの全ては  $Cat(A)$  と表記する (Categori patterns)。それらのパターンは直接に登録されたパターンと結合したパターンである。

$$Cat(A) = \sum PA(i) = Com(A) + PP(A)$$

(Com: Composed patterns, PP: Primitive Patterns)

文字 A は部首 B と C からなるとすれば、結合されるパターンの組は次のように表記する

$$Com(A) = \Pi(Cat(B), Cat(C))$$

(A)

「 $\Pi$ 」の演算はカテゴリ B と C のパターンをそれぞれの一つずつを結合する。(“ $\times$ ”で表記する)

$$Cat(B) = \sum PB(i), Cat(C) = \sum PC(j)$$

i j

文字 A の順結合だけの場合は

$$Com(A) = \Pi(Cat(B), Cat(C)) = \sum (PB(i) \times PC(j))$$

(A) (i, j)

文字 A の逆結合がある場合は

$$Com(A) = \sum (PB(i) \times PC(j)) + \sum (PC(i) \times PB(j))$$

(i, j) (i, j)

(“えんじょう”と“しんじょう”がある文字のみは順結合, 逆結合の両方を行う。例：延, 建, 進, 道...)

図1に示すように、文字 A における部首 B のパターン  $PB(i)$  と部首 C のパターン  $PC(j)$  の演算 “ $\times$ ” は次のように行う。

A Method of fast handwritten character recognition without stroke count constraint.

Tu Levan and Katsuhiko Akiyama and Masaki Nakagawa

Tokyo University of Agriculture and Technology

1. パターン PB(i)を部首Bのサイズから文字Aの第一部分のサイズに変換して第一部分の左上の点に平行移動する。
2. パターン PC(i)についてPB(i)と同様にする。
3. パターン PB(i)の終点をパターンPC(j)の始点につなげる。
4. できたパターンを文字Aのサイズに変換する。

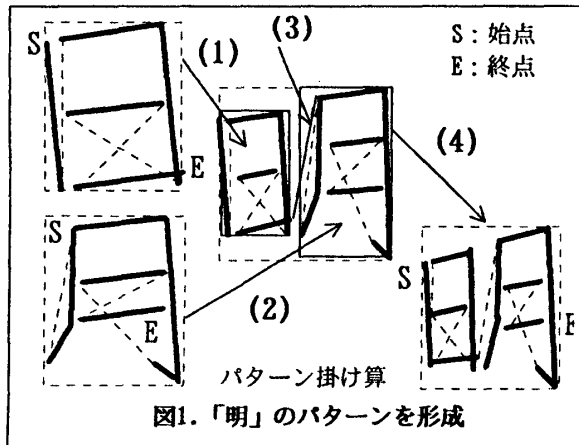


図1. 「明」のパターンを形成

図2に示すように、文字Aのパターンの数がs、部首BとCのパターンの数がm,nとしたら文字Aの全体パターン数は  $m \times n + s$  になる。ちなみに、部首BとCについて、それぞれ違う書き順パターンが数十あるとすると、文字Aの違う書き順のパターン数が数百になる。

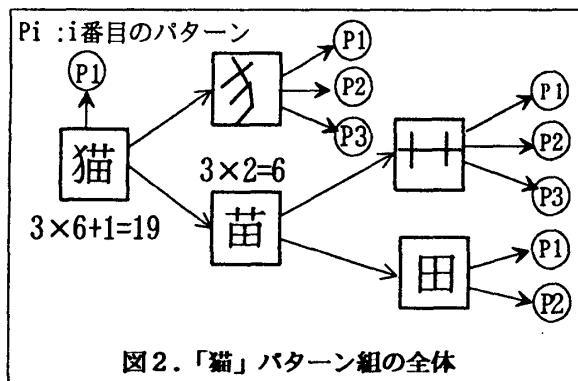


図2. 「猫」パターン組の全体

けもの偏のパターン数：3個

くさ冠のパターン数：3個

「田」のパターン数：2個

(それぞれ異なる書き順のパターン)

「猫」は直接に一つのパターンが登録されており、全体のパターンを結合するとそれぞれ書き順の異なる19個のパターンができる。

4. 実現と評価：

上に述べた構成で、45人分のデータ(約15万パターン)を観察して典型的な書き順のパターンを辞書に入れて、サイズ300KB程度の汎用な辞書を

作った。観察に用いていない8人分のデータ(10712文字\*8人分)を認識させ評価した。このデータは筆者に続け字や書き順等を制限せずに採集したものであり、誤字の訂正などを行っていないものである。評価結果は次の通りである。  
文字範囲：漢字2010, 仮名160, アルファベット・記号82  
機械：DELL 4100/MXV 環境：Windows  
平均時間：0.45秒/字(大分類なし)

表.1 データの評価(文脈処理なし/単位%)

データ	第一	第二	第三	第四	第五
A	90.10	95.74	96.77	97.32	97.58
B	83.51	92.23	93.73	94.41	94.71
C	87.48	94.17	95.93	96.67	96.95
D	91.18	96.28	97.26	97.60	97.76
E	90.54	96.12	97.46	97.95	98.13
F	84.07	91.45	93.59	94.58	95.11
G	90.08	95.08	96.60	97.09	97.32
H	81.93	89.58	91.76	92.83	93.51
平均	87.36	93.83	95.39	96.06	96.38

この実験の結果、一般的でない書き順や激しい字体のくずれ(人間が読めないくらい)が少ないデータ場合、第3候補までに正しい文字が97%程度が含まれると考えられる。現在文脈処理との試験的な結合により平均94%程度の認識率を得ている。

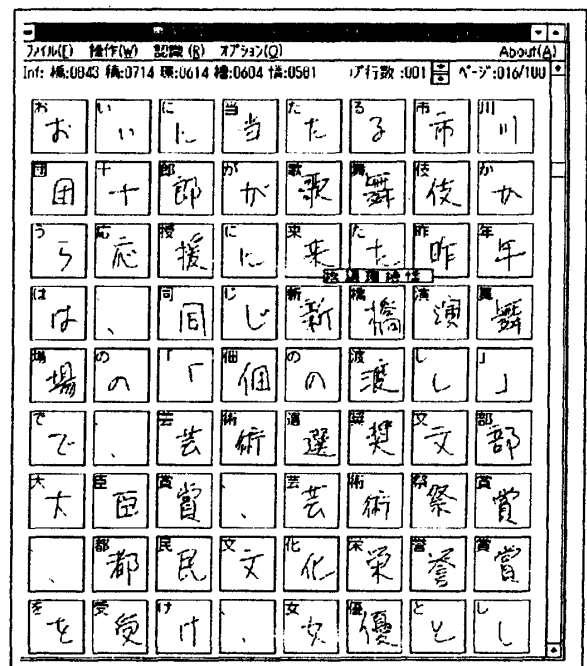


図3. 認識結果の例