

候補文字ラティス法による枠無し筆記文字列の オンライン認識

正員 村瀬 洋[†] 正員 若原 徹[†] 正員 梅田三千雄[†]

Online Writing-Box Free Character String Recognition by
Candidate Character Lattice Method

Hiroshi MURASE[†], Toru WAKAHARA[†] and Michio UMEDA[†], *Members*

あらまし 筆記制限の少ない文字認識技術は真に扱い易い入力形態を追求するうえで重要である。本論文では、セグメンテーション指示なく野線上に自由筆記された仮名漢字文字列のオンライン認識法として、以下の4段階から構成される「候補文字ラティス法」を提案する。①入力文字列中から文字らしい部分を候補文字として抽出する。②候補文字を認識し、結果を相異度とともに候補文字ラティスに登録する。③ラティス探索により相異度の和を最小とする文字系列を認識結果として選択する。④偏と旁により左右に分離する文字のセグメンテーションに対し文字の縦横比を利用した分離有意文字処理を適用し認識結果を修正する。上記処理により、文字列からの文字のセグメンテーションと各文字の認識とが融合され、不定ピッチ文字列、文字間で重なる文字列、大小文字混在の文字列等を高精度に認識することが可能になる。認識実験を行った結果、105個の文字列に対し96.3%の認識率が得られ、本手法の有効性が確認された。

1. ま え が き

漢字情報の簡単な入力法として文字認識技術の活用が期待されている。その際必要となる技術に低品質文字認識⁽¹⁾、枠無し筆記文字列認識等があり、筆記制限の緩和が課題となる。本論文で述べるオンライン枠無し筆記文字列の認識は、即時性や対話性が良好であるというオンライン認識の特長に加え、枠内筆記の制限の解除により精神的負担が大きく低減されるという特長を持つ。また、この認識技術は図面中の文字のような任意の場所に大小混在した文字や、野線上に自由筆記された文字の認識等広い応用範囲を持つ。

従来、枠無し筆記文字列を認識する手法は、文字と文字を分離するセグメンテーション段階と、個々の文字を認識する文字認識段階から構成されていた。セグメンテーション手法には、文字と文字の間でピック等の指示を与える手法や、1文字を書き終えてから次の文字を書き始めるまでの時間間隔を利用する手法⁽²⁾があるが、いずれも筆記者の負担が大きくなる。一方、

英文字列を対象にストロークの横軸への投影を利用して自動的にセグメンテーションを行う手法⁽²⁾（以下、単純投影法と呼ぶ）等もあるが、漢字のように左右・上下に分離する可能性の高い対象では、ストロークの位置関係の利用だけで文字の正確なセグメンテーションを得ることは困難である。この他にOCR型文字列を対象とした報告^{(3)~(5)}があるが、これらの手法をそのままオンライン型の文字列に適用することは不適當である。他方、文字認識には多数の手法^{(6)~(8)}が提案されているが、文字が正しくセグメンテーションされなければ、これら文字認識技術は本来適用出来ない。

本論文では、タブレット上に枠無し筆記された仮名漢字文字列を認識するため、各文字のセグメンテーションと各文字の認識とを融合した「候補文字ラティス法」を提案する。本手法は筆者らの提案による線図形認識法である「候補ラティス法」⁽⁹⁾を文字認識用に改良したものであり、音声認識で知られる「音韻ラティス」⁽¹⁰⁾と同様の発想に基づく。本手法の特徴を以下に述べる。

- (1) 本方式では本質的に文字間隔（文字ピッチ）の情報を使用しないため、手書きによる文字間隔の変動には認識精度は影響されない。

[†]日本電信電話株式会社武蔵野電気通信研究所, 武蔵野市
Musashino Electrical Communication Laboratory, NTT,
Musashino-shi, 180 Japan

- (2) 漢字や平仮名には、“科”，“川”，“い”等の単純処理では部分図形に分離してしまう文字が多数存在する．本手法では各文字の認識情報が自動的に文字のセグメンテーションにも利用されるため，分離した部分図形が他の文字にならなければ1文字として正しく認識される．
- (3) 漢字には，“理”，“炉”等のように左右に分離し，分離した各部分図形が他の文字になる場合がある．この場合には，文字の縦横比を利用した分離有意文字処理を導入している．

2. 認識対象

日本語文書には縦書きと横書きが存在するが，横書き文字列の認識技術が確立されればこの技術を縦書き文字列に拡張することは容易であるため，認識対象として横書き文字列を扱う．文字列は常用漢字1945字種と平仮名46字種の文字から構成されるものとする．

2.1 文字列データ

本論文で述べる認識手法の効果を確認するため，文字列データを以下の条件で収集した．

- (1) 横罫線(1 cm間隔)の用紙に，文字列を自由なピッチで筆記させた．
- (2) 各文字の筆順は自由であるが，画数は守らせた．
- (3) 文字列として以下の5文例を用意し，それを筆記させた．
 - ① 地理的に離れた場所での
 - ② 詳細な検討を続ける
 - ③ 高騒音下での通話対策
 - ④ 安定でしかも量産に向く
 - ⑤ 利用するための手引
- (4) 文字ピッチについては密・普通・粗の3種類を指示し，同一文例を3回づつ筆記させた．

データ収集数は(7名，各5文例，各3回)の105文例であり，文字総数は1050文字である．収集した文字列の一例を図1に示す．

2.2 辞書作成データ

文字認識に用いる標準パターンの作成には，常用漢字1945字種，平仮名46字種，画数変動文字126字種(例えば，“子”は正しくは3画であるが2画で筆

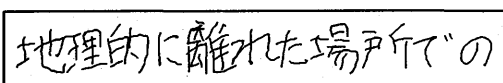


図1 文字列の一例
Fig. 1 An example of character string.

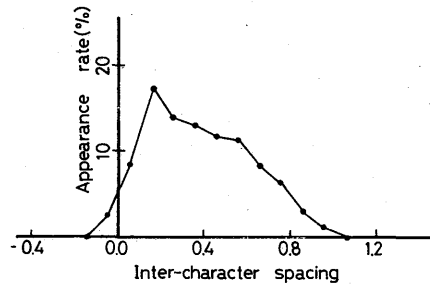


図2 文字間隔に対する出現頻度
Fig. 2 Appearance rate of inter-character spacing.

記されやすい．このような文字)の合計2148字種20名分(42960文字)を使用した．

3. 手書き文字列の性質

手書き文字列の性質を予備実験として調査する．

3.1 文字間隔の分布

手書き文字列の文字間隔の変動は印刷文字の場合と異なり極めて大きい．図2に2.1で述べた文字列データについて，X軸(横方向)に着目した文字間隔の分布を示す．但し，文字間隔の値は，文字列の縦幅を1.0とするように大きさを正規化(4.2参照)した後の値である．負の文字間隔は文字が重なり合うことを意味する．図2より，手書きでは文字が10%程度重なる場合から，文字間が1文字分以上離れる場合まで存在することが確認された．この事実は，手書き文字列のセグメンテーションに単純投影法の適用が困難であることを示す．

3.2 分離文字の存在

漢字や平仮名には，“科”，“川”，“い”等のように，左右に分離する文字が多数存在する．そこで，辞書作成データ(2.2参照)をもとに単独文字の分離度を調査した．その結果，文字のX軸への投影が分離する文字は43%，文字の高さに比較し10%以上投影が分離する文字は15%存在することを確認した．この事実は，文字のセグメンテーションに単純投影法を適用したならば，単独文字が複数の図形に分離される可能性が高いことを示す．

4. 認識アルゴリズム(候補文字ラティス法)

本論文で提案する候補文字ラティス法は文字のセグメンテーションと認識を同時に実現することにより，セグメンテーション精度を高める手法である．処理は，

文字列から基本セグメントへの分解, 基本セグメントから候補文字の生成, 候補文字の認識とその結果の候補文字ラティスへの登録, 候補文字ラティスの探索, 分離有意文字処理の各段階から構成される。以下, 各処理を詳細に述べる。

4.1 手書き文字列入力

手書き情報の入力にはデータタブレットを使用する。タブレットはペンのアップダウン情報とXY座標の値を一定時間間隔でサンプリングする機能を有する。ペンのダウンからアップまでの座標値系列をストロークと呼ぶことにする。

4.2 前処理

前処理として, 文字列全体の大きさの正規化とストローク毎の座標点数の規格化を行う。大きさは, 各座標点のY座標の2次モーメントの値が一定になるように正規化する。一方, 入力された座標点は一定時間間隔でサンプリングされたものであるため, 引き続き座標点間の距離は筆記速度により変動する。そこで引き続き座標点間の距離が一定値となるようにストロークを構成する座標系列を変換する⁽¹⁾。

4.3 基本セグメントへの分割

文字列を構成する入力ストローク列(ストローク数をMとする)を複数の基本セグメントに分割する。方針は, 文字間の分離箇所が含まれない範囲で, なるべく大きなストロークの組を基本セグメントとして分割することにある。基本セグメントの分割パラメータ T_h を以下のように導入する。 $1 \leq k < M$ のすべてのkに対して, 第1ストロークから第kストロークまでのX座標の最大値を X_1 , 第k+1ストロークから第MストロークまでのX座標の最小値を X_2 とし, $(X_2 - X_1) > T_h$ を満足するkをすべて求める。各kに対し第kストロークと第k+1ストロークとの間で入力ストローク列を分割し, 分割された各ストロークの組を基本セグメントとする。図3に, T_h に対する基本セグメントの例を示す。 T_h を小さくすれば基本セグメントの単位が小さくなるが, T_h を大きくしすぎれば基本セグメント内に文字間の分離箇所が含まれる。図2より, 通常横方向に手書きされる文字列では文字間の重さなりは15%以下と考えられる。この結果より, T_h として-0.15を設定する。分割後の基本セグメントを e_1, \dots, e_t とする。但し, tは基本セグメント総数である。

4.4 候補文字の生成

文字列中の各文字は基本セグメント(e_1, \dots, e_t)を

組合わせることにより構成されるが, この段階では組合わせ方は不明である。そこで, 基本セグメントを組合わせ, 組合わせた図形が文字らしくなるすべての場合に対し, この図形を候補文字とする。基本セグメントを組合わせた図形が候補文字となるための条件として,

- ① 図形は引き続き基本セグメントから構成される。
- ② 図形の横幅は2.0以下である。
- ③ 図形を囲む長方形の長辺は0.2以上である。
- ④ 図形のストローク数は23以下(常用漢字の最大画数は23)である。

と設定した。但し, ②③で設定したいき値は, 文字列に極端に縦横変形した文字や非常に小さな文字を含まない限り十分な値であり, 今回の文字列データで

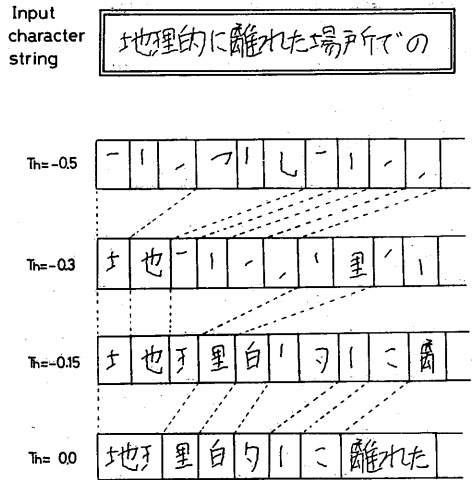


図3 T_h に対する基本セグメントの例
Fig. 3 Examples of basic segments vs. T_h .

Basic segment	e1	e2	e3	e4	e5	e6	e7
	ち	也	牙	り	里	白	り
	ち	也	牙	り	里	白	り
	地	牙	白				
	也	里	白				
	地		白				
	也	白					
		理	的				

図4 候補文字の生成
Fig. 4 Candidate character generation.

もすべての文字についてこのいき値を満足していた。図3で示した基本セグメント ($T_h = -0.15$) から生成した候補文字を図4に示す。候補文字数は実際の文字数に比較して一般に多数生成される。図4に示した例では、11文字から構成される文字列に対し、24の基本セグメントが抽出され、基本セグメントの組み合わせにより98個の候補文字が生成される。

4.5 候補文字の認識

前節で生成された各候補文字を認識することにより、各候補文字はどの文字らしいか(候補文字認識結果)、どの程度その文字らしいか(距離値)を求める。以下候補文字の認識処理を詳細に述べる。

(1) 前処理

前処理として、候補文字の座標値系列の重心と重心回りの2次モーメントを利用して、候補文字の位置と大きさの正規化を行う。

(2) 候補文字認識用の辞書の作成

画数変動文字を含んだ常用漢字と平仮名2148カテゴリーについて辞書を作成する。最初に、辞書作成用文字の各ストロークの座標系列を一定の点数(ここでは15点)で近似し、前処理として上記(1)の処理を施す。次に各カテゴリー毎に20個の文字を使用し、対応する座標点毎に座標値の平均値を求める。最後に、座標点間の距離が一定値になるように変換する(4.2の後半の処理を参照)。この座標値系列を候補文字認識用の辞書とする。

(3) ストローク間の距離の定義(位置と方向のDPマッチング)

ストローク間の距離は(4)で述べる文字間の距離の計算に使用される。2本のストローク間の距離をストローク各点の位置(座標値)とその点での接線方向の値のDPマッチング⁽⁹⁾により定義する。DPマッチングは低品質文字認識に有効である^{(1), (6)}。一方のストロークの座標値系列を $\{(U_i, V_i) \mid i=1, \dots, I\}$ とし、他方を $\{(U'_j, V'_j) \mid j=1, \dots, J\}$ とする場合、両者のストローク間距離 d を、

$$d = \frac{1}{J} \min_u \left\{ \sum_{j=1}^J (g(j, u(j)) + \alpha \cdot h(j, u(j))) \right\} \quad (1)$$

で定義する。但し、 I, J はストロークの近似点数であり、 I と J とは一般に異なる。 $g(i, j)$ は (U_i, V_i) と (U'_j, V'_j) の間のユークリッド距離、 $h(i, j)$ は、点 (U_i, V_i) から点 (U_{i+1}, V_{i+1}) に向かうベクトルと、点 (U'_j, V'_j) から点 (U'_{j+1}, V'_{j+1}) に向かうベクトルのなす角度で定義する。 $u(j)$ は座標点間の対応関係を表

わす関数であるが、不適当な対応を防止するため制限条件として、

$$\begin{cases} u(1) = 1 & (2) \\ u(J) = I & (3) \\ u(j) = i \text{ のとき} & \\ \quad u(j+1) = \{i \text{ or } i+1 \text{ or } i+2\} & (4) \end{cases}$$

を課した。 α は接線方向の特徴の寄与度をあらかじめ定数であり、予備実験により最適値に設定した。

(4) 文字間距離の計算

候補文字は複数のストロークから構成される。まず、辞書の中から候補文字のストローク数 (L とする) と同じストローク数の標準文字を選択し、選択された標準文字のすべてと候補文字との間で文字間の距離を求める。文字間の距離は対応するストローク間の距離(3参照)の和で定義する。但し、候補文字の筆順は自由であるため、ストロークの対応は最初から決定できない。そこで、ストローク間距離行列⁽⁶⁾を用いて、対応するストロークを決定する。標準文字の第 i ($1 \leq i \leq L$) ストロークと、候補文字の第 j ($1 \leq j \leq L$) ストロークとのストローク間距離を d_{ij} とすれば、ストローク間距離行列は、

$$\begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1L} \\ d_{21} & d_{22} & \cdots & d_{2L} \\ \vdots & \vdots & d_{ij} & \vdots \\ d_{L1} & d_{L2} & \cdots & d_{LL} \end{bmatrix}$$

となる。この行列の各行から最小の値を選択し(対応するストロークの決定)、それらの和を文字間の距離とする。つまり、文字間の距離 D を、

$$D = \sum_{i=1}^L \left\{ \min_j (d_{ij}) \right\} \quad (5)$$

で定義する。

(5) 候補文字の認識

候補文字と、候補文字に等しいストローク数を持つすべての標準文字(辞書)との間で文字間距離を求め、最小の距離値を持つ標準文字のコードを認識結果とする。

4.6 候補文字ラティスの作成

候補文字ラティスは、入力文字列中の各基本セグメントがどの候補文字の一部であるかの情報、各候補文字の認識結果、及び認識時の距離値を、表形式に表現したものである。図4に示した表の各欄に候補文字の認識結果と距離値を代入することにより、候補文字ラティスが生成される。候補文字ラティスの例を図5に示す。

4.7 候補文字ラティスの探索

入力文字列中に含まれる各文字のセグメンテーション

Basic segment	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10
	士	也	王	く	耳	白	く	又	し	こ
	0.72	1.01	1.61	0.39	1.98	1.16	0.47	0.94	0.46	0.56
	地		げ		略		勺		に	
	1.22		1.89		3.14		0.74		0.87	
candidate character		作		里		自		孔		
		2.73		1.37		1.57		1.23		
distance		班		軸		双				
		2.40		2.66		1.78				
		性		軸		加				
		3.40		3.27		2.07				
		理		的						
		1.85		1.76						
	理			離						
	3.70			4.30						

図5 候補文字ラティスの例
Fig. 5 An example of candidate character lattice.

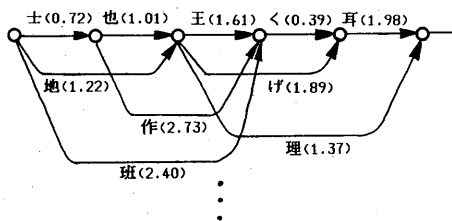


図6 候補文字ラティスのグラフ表現
Fig. 6 Graph representation of candidate character lattice.

ンと各文字の認識とを同時に行うことは、候補文字ラティスの中からすべての基本セグメントを被り最適な文字系列を選択することにより実現できる。候補文字ラティスは2端子の重み付きグラフ（候補文字の認識結果をブランチに、距離値をその重みとするグラフ）により表現できる。図5に示す候補文字ラティスに対するグラフ表現の例を図6に示す。すべての基本セグメントを被り文字系列の選択は、このグラフの一方の端子から他方の端子に向かうパスの選択に対応する。このパスは一般に多数存在する。そこで、評価関数 S を導入し、 S を最小にするパスを探索する。パスの探索には動的計画法を採用した。 S は、

$$S = \sum_{\text{ブランチ} \in \text{パス}} (\text{ブランチの重み}) \quad (6)$$

で定義する。このパスに対応する文字系列を仮の認識結果とする。図5中に、太枠で仮の認識結果を示す。

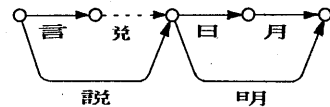


図7 “説明”の場合のグラフ探索
Fig. 7 Lattice search in case of “説明”.

4.8 分離有意文字処理

4.7までの処理により、たとえ文字が偏と旁、或は手書き変形のため左右分離した場合でも、左右の部分図形の少なくとも一方が文字として意味を持たなければ文字列は正しく認識できる。しかし、文字が左右に分離し、且つ分離した各部分図形がどちらも文字として意味を有する場合（この文字を分離有意文字と呼ぶ）には4.7までの処理では必ずしも正しく認識されない。例えば、“説明”という文字列が、“言”、“兑”、“日”、“月”の4個の基本セグメントに分割された場合には、図7のような2端子グラフが生成される。この場合には、“兑”という文字が存在しないために“説”は正しく認識されるが、分離有意文字である“明”は、“明”と認識されるか“日”+“月”と認識されるかは距離値情報だけで正しく決定できない。分離有意文字は常用漢字1945字種中266字種存在する。表1に、その一例を示す。

ある分離有意文字 F を $F = f_l + f_r$ と表現する（例えば、 $F = \text{“明”}$ 、 $f_l = \text{“日”}$ 、 $f_r = \text{“月”}$ ）。候補文字ラティス中に F とそれに対応して f_l 、 f_r なる候補

表1 分離有意文字例

F	f_l	f_r	F	f_l	f_r
暗	日	音	脂	月	旨
姻	女	因	討	言	寸
横	木	黄	軸	車	由
好	女	子	終	糸	冬
動	重	力			
鉢	金	本			

表2 認識率と誤読内訳

		分離有意文字処理	
		無し	有り
認識率		95.6%	96.3%
誤読	基本セグメント分割誤り	0.6	0.6
	ラティス探索誤り	2.8	2.1
	文字認識誤り	1.0	1.0

文字が存在し、且つ仮の認識結果(4.7参照)として F または f_l , f_r なる文字列が選択された場合には、正しく認識されていない可能性がある。そこで、この場合には文字 F の縦横比と、文字 f_l と f_r の縦横比の平均値を比較し、その値が1.0に近い方の文字(文字列)に修正する。この処理を分離有意文字処理と呼ぶ。

分離有意文字処理を経て選択された候補文字の認識結果の系列を、入力文字列の最終認識結果とする。

5. 認識実験結果

4.で述べた認識手法の効果を確かめるため、認識対象の一例として2.1で述べた105個の文字列データを使用して認識実験を行った。

5.1 分離有意文字処理を行わない場合

候補文字ラティス中から単純に距離値の和を最小とする文字系列を選択(4.1~4.7参照)することにより文字列データを認識した場合、認識率は95.6%であった。但し、認識率は文字列に含まれる文字総数に対する正しく認識された文字数の割合で定義した。表2に誤読4.4%の内訳を示す。

5.2 分離有意文字処理を行った場合

5.1に示した誤読中、ラティス探索誤りの一部は分離有意文字処理を導入することにより改善される。その結果、認識率は96.3%に向上した。表2に誤読3.7%の内訳を、図8に誤読例を示す。

input samples	recognition results	
細い検討	細に検討	糸? 討
地理的に	地理的に	地理白夕に
しかも	しかも	し力しも
利用する	利用する	利申する

図8 誤読例

Fig. 8 Misrecognized samples.

- (1)

日本

- (2)

日本 電電 公社

- (3)

東京都 武蔵野市

図9 可読文字列例

Fig. 9 Correctly recognized character string.

6. 考察

6.1 本手法の適用範囲

本手法は文字のセグメンテーションに文字ピッチや文字の大きさ等の情報を使用していないため、以下のような文字列を認識することが可能である。

- ① 文字の大きさに比較して T_h (4.3参照) までの文字間の重なりを許容する(図9(1))。
- ② 完全な不定ピッチを許容する(図9(2))。
- ③ 同一文字列内に文字の大小混在を許容する(図9(3))。

6.2 計算量

本手法では、基本セグメントの組み合わせから生成されるすべての候補文字を認識し、セグメンテーションを高精度化しているため、候補文字認識回数が多くなり、計算量は増大する。例えば、図1に示した文字列の例では、1文字に対し必要となる平均的な候補文字認識回数は約9回、そのための認識時間は1 MIPSのミニコン(言語としてFORTRAN77使用)でシミュレーションしたところ約18秒であった。必要な認

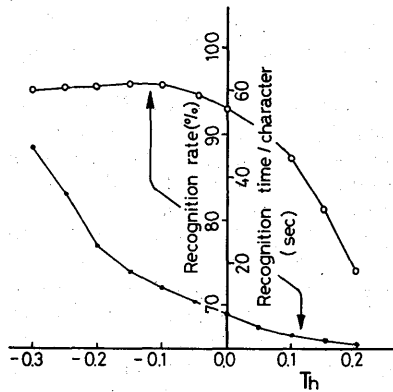


図10 認識時間/文字, 認識率 vs. T_h
Fig. 10 Recognition time/character and recognition rate vs. T_h .

認識時間は文字の重なり許容度 T_h に依存する。その結果を認識率とともに図10に示す。今回は T_h を -0.15 に設定(4.3参照)したが、この値は認識率を下げない範囲で最少の計算時間となっている。文字間の重なり更に大きい低品質な文字列を認識する為には、 T_h を小さくする必要があり、この場合認識時間は増加する。

6.3 他手法との比較

単純投影法により文字列から文字をセグメンテーションし、各文字を文字認識する手法と、本手法とを比較した。単純投影法は文字列を構成する各ストロークを横軸に投影し、その影の間隔が T (いき値) 以上になる箇所を文字をセグメンテーションする手法である。文字列データ105文例に対し、 T をパラメータとしてセグメンテーションの実験を行った。 T を小さな値に設定すれば1文字が複数の図形に分離されるため誤読は増加し、 T を大きな値に設定すれば文字間が連結するため誤読は増加する。 T を最適に設定することにより68.5%のセグメンテーション成功率が得られた。これを文字認識したところ、文字列の認識率は68.0%であった。表3に、本手法との認識率の比較を示す。

6.4 分離有意文字処理の拡張

本手法では分離有意文字対策として文字の縦横比を利用したが、これ以外に単語知識に基づく文字間の接続関係を利用することもできる。例えば、“動”は左右分離しそれが接続して“無重力”等の単語を作るが、“欲”は左右分離すると単語を作らないため“谷欠”と認識するより“欲”と認識すべきである。学研国語大辞典の見出し語約8万単語により調査した結果、分

表3 文字列の認識率

単純投影法	候補文字ラティス法
68.0%	96.3%

離有意文字266字種中251字種に対しては左右分離した文字の接続を含む単語はない。この場合には左右に分離しないと判定すれば良い。しかし、固有名詞や2単語が連続する場合をも考慮すると、上記251字種に対しても左右の文字が接続する可能性があり、上記の処理だけでは不十分である。これら分離有意文字処理の拡張は今後の検討課題とする。

7. むすび

本論文では、真に扱い易い入力形態を追求するため、セグメンテーション指示なく横方向に手書きされた枠無し筆記文字列の認識手法として「候補文字ラティス法」を考案し、その性能を実験的に確認した。以下に、まとめる。

- (1) 文字列からの文字のセグメンテーションと各文字の認識とを融合することにより、文字のピッチ等の情報を全く使用することのない高精度な文字列認識法を実現した。その結果、完全な不定ピッチ文字列、文字間の重なりを許容する文字列、大小文字混在の文字列の認識が可能となった。文字列データ105文例に対し96.3%の認識率を得た。
- (2) 文字の縦横比を利用した分離有意文字処理を導入したことにより、“理”、“炉”等の様に文字左右に分離しそれぞれが意味を持つ分離有意文字のセグメンテーション精度は向上し、認識率は導入しない際の95.6%に比較して0.7%改善された。

本論文で提案した「候補文字ラティス法」は枠無し筆記文字列のオンライン認識に極めて有効である見通しを得た。今後の課題には、更に多数の文字列データで本手法の能力を定量的に分析すること、言語知識等の上位概念を導入し分離有意文字処理を更に充実させること、本手法を函数変動を含む文字列の認識に拡張すること、等がある。

謝辞 日頃御指導頂く畔柳情報通信基礎研究部長、増田第三研究室長に深謝します。研究を進める上で有益な議論をしデータ収集に協力して頂いた第三研究室および視覚情報処理研究グループの諸氏に感謝致します。

文 献

- (1) 若原, 梅田: “ストローク結合情報を用いたオンラインくずし字分類法”, 信学技報, PRL83-30(1983).
- (2) G.F. Groner: “Realtime recognition of handprinted text”, AFIPS, FJCC, 29, p. 591 (1966).
- (3) 馬場口, 塚本, 相原: “手書き文字列における文字切り出しの基礎実験”, 信学技報, PRL83-58 (1983).
- (4) T. Akiyama, S. Naito and I. Masuda: “A method of character extraction from format-unknown document images”, ICTP, p. 85 (1983).
- (5) R.G. Casey and G. Nagy: “Recursive Segmentation and Classification of Composite Character Patterns”, Proc. of 6-th ICPR.
- (6) 吉田, 迫江: “スタックDPマッチングによるオンライン手書き文字認識”, 信学システム部門全大, 90 (1983).
- (7) M. Nakagawa, T. Manabe, K. Aoki, Y. Ikeda and N. Takahashi: “Online handwritten character recognition as Japanese input method”, ICTP, p. 191 (1983).
- (8) 小高, 若原, 増田: “筆順に依存しないオンライン手書き文字認識アルゴリズム”, 信学論(D), J65-D, 6, p. 679 (1982).
- (9) 村瀬, 若原, 梅田: “候補ラティス法による手書きフローチャートのオンライン認識”, 信学論(D), J66-D, 6 (1983).
- (10) 鹿野, 好田: “会話音声の機械認識における言語処理”, 信学論(D), J61-D, 4 (1978).

(昭和59年8月29日受付)