

# 手書き文字列読み取りのための 単語連鎖制約に基づく効率的探索と棄却

福島 俊一† 下村 秀樹† 山内 俊史‡

NEC 情報メディア研究所† 産業オートメーション事業部‡

## 1 はじめに

自由に書かれた手書き文字列は、字形が多様で、文字サイズ・文字ピッチにばらつきがあり、文字の接触や入り組みなどもよく起きる。そのため、画像的な特徴のみに頼ったパタン処理では、文字切り出し時の曖昧性を正確に絞り込めないケースや、正解文字が個別文字認識の候補中から欠落するケースが頻繁に生じる。したがって、高精度な読み取りを実現するためには、文字列を単語の連鎖としてとらえる知識処理を組み合わせることが不可欠である。

パタン処理と知識処理との組み合わせについては、直列型[1][2]、フィードバック型[3][4][5]、トップダウン型[6]、検証型[7]などのアプローチが試みられている(図1参照)。しかし、従来研究においてはまだ、前述のような手書き文字列の読み取りで発生する曖昧性や誤認識に対して十分に頑健かつ効率的な処理が実現されているとはいえない。

本論文では、検証型のアプローチを基本に、知識処理における効率的な探索能力を高め、総合的な棄却判定と多種の検証機能を実装した手書き文字列読み取り方式を提案する。さらに、本方式を郵便物の手書き宛名住所の地名領域の読み取りに適用し、その有効性を評価・実証する。

## 2 アプローチ

手書き文字列読み取りにおける大きな課題を2つあげ、それに対する本論文でのアプローチを述べる。

図2は手書き文字列の例(「富士市富士見台」)である。この画像では、文字の接触と入り組みが発生しているため、単純に文字列に垂直な方向への画像の投影を頼りに文字を切り出すと、(1,1) (2,1) (3,1) (4,4) (8,1) (9,1) (10,2)が切り出されやすい( $(t,w)$ は先頭位置が $t$ で幅が $w$ のセグメントを表す)。しかし、これらのセグメントのうちで正しく1文字が切り出されているのは(3,1)と(9,1)の2つしかない。文字切り出しの段階で、より多くの正解セグメントを得るためには、より多くの切り出し可能性を出力することになる。その結果、図

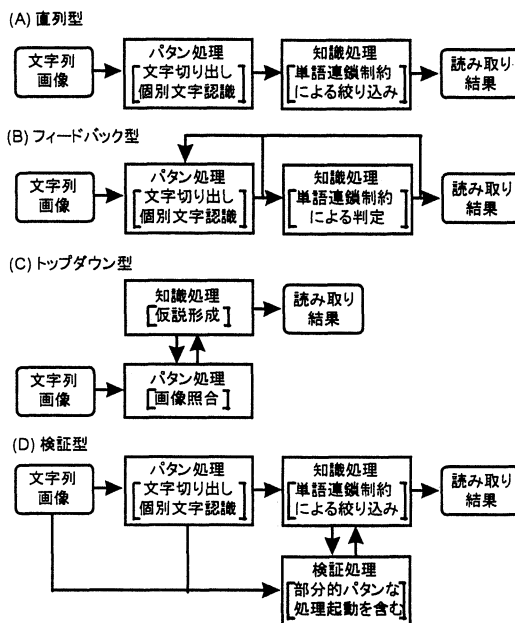


図1: パタン処理と知識処理との組み合わせ

2では、7文字に対して21種類のセグメント候補が発生しており、それらを先頭から末尾まで組み合わせるとセグメント列の種類は55通りにもなる。

従来提案されている知識処理方式(特に前述の直列型やフィードバック型の研究)では、個別文字認識で発生する曖昧性を単語連鎖制約に照らして絞り込む役割が中心で、文字切り出しの曖昧性は別に分けて扱われている。したがって、図2のように多数の切り出し可能性が発生する場合には、各可能性(この例では55通り)に対して知識処理をやり直すことになり、処理効率が大きく低下する。さらに、図2の例では、正解セグメントの欠落や正解文字の欠落が発生しており、虫食い照合や文字数の合わない照合なども必要になる。

このような第一の課題に対して筆者らは、文字を基本単位としてタグを付与し、その位置関係をチェックしながら動的計画法を用いて探索するアルゴリズム(文字タグ法[8])を開発した。文字タグ法によれば、文字切り出しにおける曖昧性を個別文字認識の曖昧性と

1	(1, 1) 字, 宇, 0	(6, 2) 澄, 港, 字, 雀, 宇
2	(2, 1) 四, 日	(6, 3) 富, 宮
3	(3, 1) 三, 土	(7, 1) 後, 陸, 焼, 彼, 佐
4	(3, 2) 色, 土	(7, 2) 遣, 道, 澄, 雀, 蓋
5	(3, 3) 茅, 嘉, 秦, 夷, 妻	(8, 1) 四
6	(4, 1) 一	(8, 2) 要, 葉, 幸, 望, 聖
7	(4, 2) 和, 市	(9, 1) 土, 七
8	(4, 3) 桑, 士	(9, 2) 夏, 養, 真, 裏, 豪
9	(5, 1) 印, ヤ, 市, 私	(10, 1) 泉, 尻
10	(5, 2) 字, 士	(11, 1) 台
11	(6, 1) 力, 門, 一, 火	

図 2: 手書き文字列とパターン処理結果の例

時に扱いながら、誤切り出しや誤認識が含まれていても効率よく最良解を探索することが可能となる。

第二の課題はパターン処理と知識処理とを連携させた適切な棄却の実現である。効率のよい探索アルゴリズムによって、パターン処理の段階で発生する多数の曖昧性を扱うことが可能になり、そのなかに正解候補が含まれる率は高まる。しかし、一方で、正解でなくとも偶然に単語連鎖制約を満たすような組み合わせができる可能性も高まる。そのようなケースで正解候補を選別し、誤り候補を棄却するためには、パターン処理から別の可能性も含めたより深い情報を取り出す必要がある。

この観点から図1のアプローチを比較すると、直列型では、最初から可能性をすべて網羅し、あらゆる情報を後段へ送り出す方針をとらざるを得ず、処理量・情報量(後段で使われない情報も含む)が膨大になってしまう。フィードバック型では、必ずしも正解ではない局所最良解に陥る危険性をもつ。トップダウン型には、仮説の数が爆発する危険や、正しい仮説の照合が失敗したときの頑健性に欠ける問題がある。それに対して検証型は、必要な部分のみ深い情報を取り出すことが可能で効率が良い。

そこで、筆者らは検証型のアプローチを発展させる方針をとった[9]。本論文に示す検証型の読み取り方式は、次のような点で従来研究[7]よりも強力なものになっている。第一に、文字タグ法をベースとすることで、読み取り対象を従来の1単語相当文字列から単語列に拡大した。第二に、多種類のトップダウン検証手段を用意し、棄却・検証能力を高めた。

### 3 文字タグ法

一般に文字列読み取りに用いる知識は、読み取り対象文字列に出現し得る単語のリストと、それらの単語間の接続可否情報であり、図3のような単語テーブルで記述できる。図3は、読み取り対象文字列に「川崎市」「宮前区」「梶ヶ谷」「宮崎」…「有馬」が出現し得、「宮前区」には「川崎市」、「梶ヶ谷」「宮崎」…「有馬」

0:	[「川崎市」, { }]	4:	[「宮前平」, {1}]
1:	[「宮前区」, {0}]	5:	[「馬絹」, {1}]
2:	[「梶ヶ谷」, {1}]	6:	[「野川」, {1}]
3:	[「宮崎」, {1}]	7:	[「有馬」, {1}]

図 3: 単語テーブル

ヶ:	[2,2,3]	川:	[0,1,3] [6,2,2]
梶:	[2,1,3]	前:	[1,2,3] [4,2,3]
宮:	[1,1,3] [3,1,2] [4,1,3]	谷:	[2,3,3]
区:	[1,3,3]	馬:	[5,1,2] [7,2,2]
絹:	[5,2,2]	平:	[4,3,3]
崎:	[0,2,3] [3,2,2]	野:	[6,1,2]
市:	[0,3,3]	有:	[7,1,2]

図 4: 文字インデックス

には「宮前区」が前接可能であることを表わしている。

さらに、文字タグ法では、単語テーブルから機械的な変換によって作成した文字インデックスを参照する。図4は、図3の単語テーブルから作成した文字インデックスであり、例えば「崎」という文字は「川崎市」の3文字中の2文字目、あるいは「宮崎」の2文字中の2文字目になり得ることを表わしている。

図5は文字タグ法による処理例である。まず、先頭側のセグメントから順に、すべてのセグメントのすべての候補文字について、文字インデックスを検索して得られる3項情報をもとに文字タグを生成する。次に、先頭側の文字タグから順に、すべての文字タグについて、それより先頭側の文字タグで、連結可能な位置関係の条件を満たし、連結したときのコストが最良になるものを見つけて連結していく。

文字タグの連結時コストは、前側の文字タグのコストに後側の文字タグの初期コストと2つの文字タグ間の間隔にもとづくコストとを累積したものである。ただし、連結時コストが極端に悪くなる場合は文字タグの連結をカットする。また、単語列の先頭/末尾の読み飛ばし分についても補正コストを加える。動的計画法によるコスト計算と文字タグ連結により、最良コストとなった文字タグ連鎖を最終的な読み取り結果とする。

なお、アルゴリズムの正確な記述は既発表[8]を参照願いたい。

### 4 棄却と検証

パターン処理(文字切り出し/個別文字認識)の結果を入力として文字タグ法を実行した後、次のような手順で棄却判定を行なう。

まず、文字タグ法の探索結果で最良コストとなった

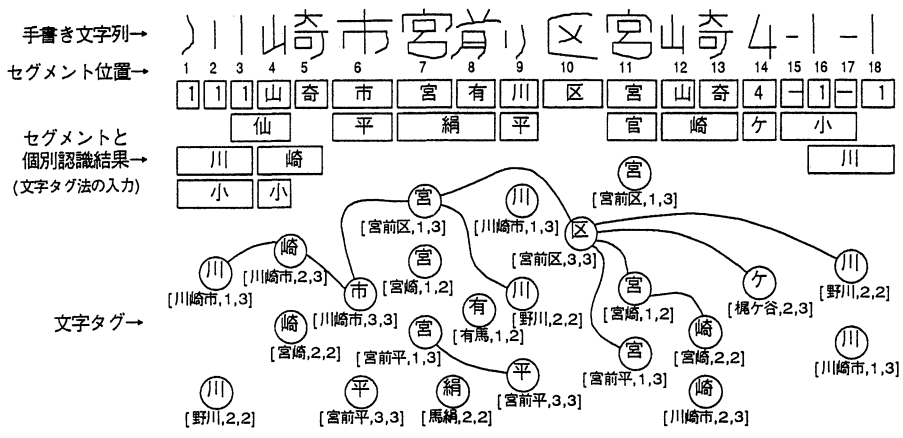


図 5: 文字タグ法による処理例

ものを基準に、一定コスト差以内かつ一定個数以内の読み取り候補を判定対象とする。次に、判定対象候補の各々について、読み取り状況に応じた尤度を計算する。この段階でトップダウン的な検証処理も実行して、尤度計算に反映させる。その結果、尤度が閾値に達していない候補は判定対象から外す。尤度が閾値に達した候補が残らない場合、および、第1位と第2位とで候補の尤度に十分な差がない場合は、読み取り結果を棄却する。そうでなければ尤度第1位の候補を読み取り結果として決定する。

トップダウン的な検証手段には、次のようなものを用意した。各検証手段の判定結果に応じて読み取り候補の尤度を加点/減点する。

**ブロックサイズ検証** 文字タグ法で読み飛ばした文字数が、対応する画像領域サイズに見合うか判定する。

**画数検証** 文字タグ法で読み飛ばした文字の画数が、対応するセグメントの文字候補群の平均画数に見合うか判定する。文字の複雑度(黒さ)を画数で近似している。

**擬似再認識** 文字タグ法で読み飛ばした文字が、対応するセグメントの個別文字認識の下位候補(文字タグ法への入力外の候補)に含まれているか調べる。

**真性再認識(1文字)** 文字タグ法で読み飛ばした文字、および、あらかじめ定義した特定文字について、文字認識方式を変えたり、文字パターンを変形させるなどして[10]、再度、個別文字認識をやり直す。

**真性再認識(2文字)** 文字タグ法の読み飛ばしが2文字の場合は、対応する画像領域を強制的に2分割して、個別文字認識をやり直す。

## 5 評価

文字タグ法の探索能力およびトップダウン的な検証を含む棄却の効果を確認するため、葉書に書かれた縦書きの手書き宛名住所データを用いて評価した。文字切り出し・個別文字認識の方式や、手書き宛名住所を想定した文字タグ法や棄却のデザインパラメータなどは、既報告[8][11][12]にしたがった。読み取り対象の地名は、東京都国分寺市の全域(セットK)と静岡県富士市の全域(セットF)の2種類を用意した。単語テーブルに登録した住所要素(市区郡名・町名など)の規模は、セットK: 49個、セットF: 231個である。また、入力画像データ総数は約800件の規模で評価を行なった。

$L$  を入力データの文字列長、 $M$  を正解文字当たりの平均候補文字数としたとき、今回のコスト定義による文字タグ法の時間計算量は  $O(L \cdot M^2)$ 、メモリ使用量は  $O(L \cdot M)$  である。実際には、地名領域以外の番地領域や氏名領域も入力に与えたので  $L = 10 \sim 30$  程度、正解文字数に対するセグメント候補数の比は平均1.9個、各セグメント当たりの候補文字数は平均2.8個(最大5、最小1)、よって  $M = 1.9 \times 2.8 = 5.3$  となった。このような入力データに対して、文字タグ法自体の実行時間は、NEC EWS4800/330 (CPU: R4400、クロック: 67MHz) で20ミリ秒未満/件で、十分高速であった。なお、図2は今回の入力データの1つであるが、この例でも文字タグ法は虫食いを含む多数の組み合わせのなかから正解を1位候補として探索できている。

読み取り精度は、正解地名文字列の構成文字が入力データ中にどれくらい含まれていたかの割合  $R$  (正解文字存在率) と関係付けて測定した。文字タグ法のみ(棄却なし)、閾値による棄却の導入、さらにトップダウ

正解文字存在率 R	処理形態	セットK			セットF		
		正読率	誤読率	棄却率	正読率	誤読率	棄却率
0% ≤ R < 40%	文字タグ法	9.1	90.9	0.0	7.7	92.3	0.0
	+ 閾値による棄却	0.0	9.1	90.9	0.0	0.0	100.0
	+ トップダウン検証	0.0	9.1	90.9	0.0	0.0	100.0
40% ≤ R < 60%	文字タグ法	41.2	58.8	0.0	25.0	75.0	0.0
	+ 閾値による棄却	17.6	23.5	58.8	12.5	3.1	84.4
	+ トップダウン検証	17.6	23.5	58.8	9.4	3.1	87.5
60% ≤ R < 80%	文字タグ法	84.5	15.5	0.0	68.0	32.0	0.0
	+ 閾値による棄却	75.9	0.0	24.1	44.0	2.7	53.5
	+ トップダウン検証	75.9	0.0	24.1	44.0	2.7	53.5
80% ≤ R < 100%	文字タグ法	89.2	10.8	0.0	93.3	6.7	0.0
	+ 閾値による棄却	82.4	0.0	17.6	61.9	1.9	36.2
	+ トップダウン検証	85.1	0.0	14.9	61.9	1.9	36.2
R = 100%	文字タグ法	100.0	0.0	0.0	95.2	4.8	0.0
	+ 閾値による棄却	98.8	0.0	1.2	98.2	0.0	1.8
	+ トップダウン検証	98.8	0.0	1.2	98.2	0.0	1.8

図 6: 読み取り精度

的な検証の導入という3段階を比較して、図6に示した。閾値棄却によって全般に良好な棄却性能が得られており、トップダウン的な検証についても、率としての貢献は目立たないが、起動された箇所では、ブロックサイズ検証、画数検証、擬似再認識とも大半がうまく作用した。検証により誤読へ変化した例としては「日の出町」を「日ノ出町」と読んだものがあるが、町名異表記を登録していたことによる。図6で誤読にカウントされているなかには、この種の異表記によるものが含まれている。その他の誤読は、誤認識文字の組み合わせが偶然に町名に読めてしまうケースであり、真性再認識に期待するところが大きい。しかし、今回実装した真性再認識は、他の検証が効かなかったときに限定して起動するようにしたことや、最初のパタン処理と方式的な差が小さい(トップダウン情報を十分に活かされてない)ことなどから、今回の規模の評価では効果が現われるに至らなかった。

## 6 おわりに

手書き文字列の頑健かつ効率的な読み取り方式を提案した。本方式の第一の特長は、文字切り出し時の曖昧性を個別文字認識の曖昧性と同時に扱い、単語連鎖制約に基づく効率のよい最良解探索を実行する知識処理アルゴリズム(文字タグ法)にある。第二の特長は、パタン処理と知識処理との連携には、必要な部分のみ深い情報を取り出す検証型のアプローチを採用し、総合的な棄却判定と多種の検証機構を実装したことである。

本方式は、郵便物の手書き宛名住所の読み取りに適用して、そのアプローチの有効性が確認できた。しか

し、棄却率と誤読率とのトレードオフを考慮した尤度計算やトップダウン的な検証の実行条件など、精度向上にかかわる要因に関して調整・検討はまだ十分ではない。今後、詳細な分析と改良を進めていきたい。

最後に、真性再認識の実現は濱中雅彦氏・石寺永記氏らによるものであり、検証型アプローチに関しては NEC 情報メディア研究所・産業オートメーション事業部の関係者から多くの示唆を受けたことを深謝する。

## 参考文献

- [1] 高尾ほか、日本語文書リーダ後処理の実現と評価、情処論:30(11)、1989年。
- [2] 伊藤ほか、OCR入力された日本語文の誤り検出と自動訂正、情処論:33(5)、1992年。
- [3] 村瀬ほか、言語情報を利用した手書き文字列からの文字切り出しと認識、信学論:J69-D(9)、1986年。
- [4] 木谷、手書き文書の文字認識結果に対する後処理方式、情処研報:91-NL-86-1、1991年。
- [5] 丹羽ほか、パターンと記号の統合化処理による文字認識、信学論:J79-D-II(2)、1995年。
- [6] 佐瀬ほか、制限付き文字列読み取りの一検討、信学技報:PRU88-115、1988年。
- [7] 仲林ほか、単語あいまい検索法を利用した枠無し文字切り出し手法、情処34 全大:4E-8、1987年。
- [8] 福島ほか、手書き文字列読み取りのための単語列探索アルゴリズム—文字タグ法—、情処研報:95-NL-107-6、1995年。(情処論:37(4)に掲載予定)
- [9] 下村ほか、手書き住所読み取りにおけるパタン処理と連携した住所知識処理方式、情処50 全大:4D-1、1995年。
- [10] 濱中ほか、変形推定を用いた検証型文字列認識の検討、信学総大:D-542、1995年。
- [11] 石寺ほか、手書き住所読み取りのための文字切り出し方法、信学総大:D-576、1995年。
- [12] 山内ほか、多段弾性照合法の郵便あて名区分機への応用、信学会情報システム大会:D-200、1995年。