

ボトムアップ／トップダウン処理を融合した 手書き文字列読み取り知識処理

下村秀樹* 福島俊一* 山内俊史**

NEC 情報メディア研究所*

NEC 産業オートメーション事業部**

本稿では、ボトムアップによる候補探索とトップダウンによる検証を組み合わせた手書き文字列読み取り知識処理方式を提案するとともに、それを住所読み取りに適用した際の評価を述べる。ボトムアップ処理では、個別文字の切り出しや認識に誤りがあっても効率よく読み取り文字列候補を探索できる「文字タグ法」によって候補抽出を行う。トップダウン処理では、認識不良に対する再文字認識や文字ブロックサイズに着目した検証等を行う。住所を対象にした実験により、処理速度と処理精度の面からの有効性が示された。

A knowledge-based processing method based on a bottom-up and top-down hybrid approach for handwritten character string reading

Hideki SHIMOMURA* Toshikazu FUKUSHIMA* Toshifumi YAMAUCHI**
Information Technology Res. Labs., NEC* Industrial Automation Div., NEC**

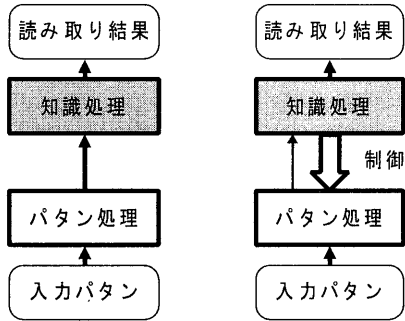
This paper proposes a knowledge-based processing method based on a bottom-up and top-down hybrid approach for handwritten character string reading, and also describes its evaluation. In its bottom-up approach, it employs an efficient string search algorithm which can extract string candidates from an error containing pattern processing result. In the top-down approach, it checks recognition error areas in the candidate strings by various verification methods. Evaluation results in applying the proposed method for handwritten address reading show its efficiency and usability.

1. はじめに

手書きや印刷の文字列を計算機で読み取ることに対するニーズは高い。しかし 100%の精度を持つ個別文字切り出しや認識の実現は考えられず、文字列読み取り技術の実用化には、言語的な知識、対象世界の知識に基づく「知識処理」との組合せが不可欠である。特に記入枠制約なしの手書き文字列の読み取りにおいては、文字ピッチの変動、文字の変形、隣接文字どうしの接触や入り組みが起りやすく、個別文字切り出し／認識（以下「パタン処理」）が非常に難しい。ときにはパタン処理出力から正解候補の一部が欠落することもあり、「知識処理」の役割は大きい。

従来の代表的な文字列読み取り知識処理方式は、図 1(a)に示されるボトムアップ型を基本としたものが多い[1][2]。すなわち、パタン処理が出力する候補群と、単語辞書や単語

間接続関係といった知識とを照合し、最も良く合致するものを結果とする直列型の構造である。しかし単なるボトムアップ型は、「パタン処理出力中の正解含有率が高くないと正しい読み取りが難しい」反面、「パタン処理の出力候補が多すぎると知識処理での処理量が爆発的に増える」といった本質的問題がある。枠なし手書きでは、パタン処理が多くの候補を出力しないと高い正解含有率を実現できず、一種の自己矛盾を抱えることになる。ボトムアップ型の拡張として、まず上位候補を仮に作成し、知識との一致度が不十分である場合に再パタン処理を行うフィードバックループを導入した方式もある[3]。しかし、一致が不十分だからといって処理を単にやり直すだけでは、効率的な再処理となるかどうか疑問である。また、フィードバック停止の判断が非常に難しい。



(a) ボトムアップ型 (b) トップダウン型

図 1 従来の代表的認識知識処理方式

一方、住所読み取りなど対象を狭く限定した場合に限って、図 1(b)のように知識からトップダウンにパターン処理を制御する方式も提案されている[4]。しかし、初期仮説の生成が非常に難しく、読み取り対象を広げると極端に精度が落ちる危険が高い。

これらトップダウンとボトムアップの長所を活かすため、これら融合する発想も提案されており[5]、我々も枠なし手書きを対象としこれに取り組んでいる[6]。概念を図 2 に示す¹⁾。以下本稿では、このアプローチを「B/T 融合型アプローチ (ボトムアップ/トップダウン融合型アプローチ)」と呼ぶ。

B/T 融合型アプローチの利点は、画像から読み取り結果を得るまでの全処理量の低減と、処理時間の適切な配分が可能になる点にある。

単なるボトムアップ型で正しい読み取りをするためには、パターン処理出力中の高い正解含有率が要求されることは前に述べた。例えば、パターン処理で文字の強制分離や文字認識候補数を増やすなどの対策が採られる。その結果、確かに正解含有率は増すが、一方でパターン処理に時間を浪費し、多くの不要な候補を生成することも事実である。また、知識処理も多くの不要候補が混じった空間を探索することになり、処理時間の増加とともに偶発的誤読の危険が増加する。

これに対し B/T 融合型アプローチでは、ボ

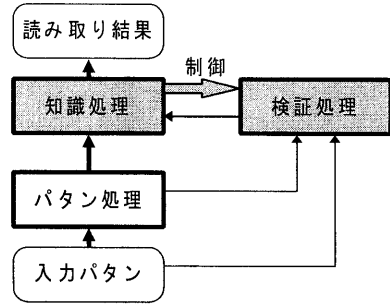


図 2 Bottom-up/Top-down 融合型アプローチ

トムアップ部分がトップダウン処理のための仮説生成と位置づけられる。つまり、仮説を生成するに十分な文字候補が出力されればよく、すべての文字切り出し/認識の可能性の考慮を必ずしも要求しない。その結果、最終的に無駄となるようなパターン処理の実行を最小限に抑えることを可能とする。

一方、トップダウン処理ではパターン処理が不完全だった部分に関して、知識から推測される文字列であるかどうかを確認する局所的詳細処理 (検証) を行う。すなわち、パターン処理を詳細に行うべき箇所を知識処理から指定してやることになり、限られ処理時間を詳細処理が必要な場所に、適切に配分することが可能となる。

このように、パターン処理と知識処理が相互に連携し合う処理方式が単なるボトムアップやトップダウンよりも利点の多いことは推測に難くない。しかし、十分な候補抽出能力を持つボトムアップ探索手法、トップダウンでの適切な検証など、この方式のメリットを十分に活かすためにはまだ課題がある。

本稿では、B/T 融合型アプローチに基づくことを前提として、ボトムアップでの効率的な候補探索方式、およびトップダウンでのパターン処理を利用した多様な検証方式を提案する。次に、それらを組み合わせた手書き住所読み取りシステムを試作し、B/T 融合型アプローチ、および提案した要素技術の効果を議論する。

*1 単なる再処理ではなく、知識に基づく推測 (「この箇所はこの文字列であるはずだ」との仮説) を確認するという点で、トップダウンと単なるフィードバックとは一線を画している。

2. 効率的ボトムアップ候補探索方式

トップダウン処理を有効に機能させるためには、パタン処理結果の多候補中から、しかも一部正解文字の欠落可能性を前提として、適切な候補を挙げる技術が求められる。これに対して我々は、「文字タグ法」を提案した[7]。ここでは、その概要と長所を簡単に説明する。

文字タグ法は主に3つのステップから成る。

[step1] 各文字切り出し結果に対する各候補文字に対して、単語辞書中のどの単語の何文字目になるか、というタグを付与する。

[step2] 付与したタグをビタビアルゴリズムで接続して候補単語列を求める。この際、途中の文字を飛ばす接続を許す。

[step3] 末端のタグから候補をたどり、尤度の高いいくつかの候補を出力する。

文字タグ法では、単語の構成情報をタグの中に巧妙に組み込んでおり、同一単語内、あるいは連鎖可能な単語間の文字タグと文字タグを接続していくことによって、尤度の高い単語列が自然に抽出できる枠組みを提供している。付与したタグの接続処理中に、切り出しの多候補と認識の多候補への対処がともに吸収されている。探索に使用する評価値は同一単語内、あるいは連鎖可能な単語間の文字を、できるだけ飛ばさずに接続したものほど

高く付ける。また飛ばしが起こったとしても、論理的に飛ばした文字数と物理的に飛ばしたブロックの数が一致しているほど高くする。この文字の飛ばしが、柔軟な不完全一致を可能としている。図3は文字タグ法を住所読み取りに適用した場合の処理イメージである。

文字タグ法の特長を次に整理する。B/T融合型アプローチの基本アルゴリズムに適した特長を持っていることがわかる。

[特長 1] パタン処理の多候補を単一レベルで扱うことができるので、処理の階層化による無駄がない。同時に、単語境界に切り出し不良があった場合に処理が特殊になるという従来問題を自然に解消している。

[特長 2] パタン処理の多候補、正解文字の欠落に対して、探索量の爆発を抑えて適切な読み取り候補を出力できる。入力文字列長に対しては、比例する時間/メモリで処理可能である。

3. 多様な検証方式の必要性[6]

パタン処理で何らかの失敗が起こった場合、パタン処理のやり直し、すなわち文字領域の再切り出し/再認識を行う方法が、従来から多く提案されている[3][5]。この際、知識による推測をトップダウン的に渡せば、なお効果的である。

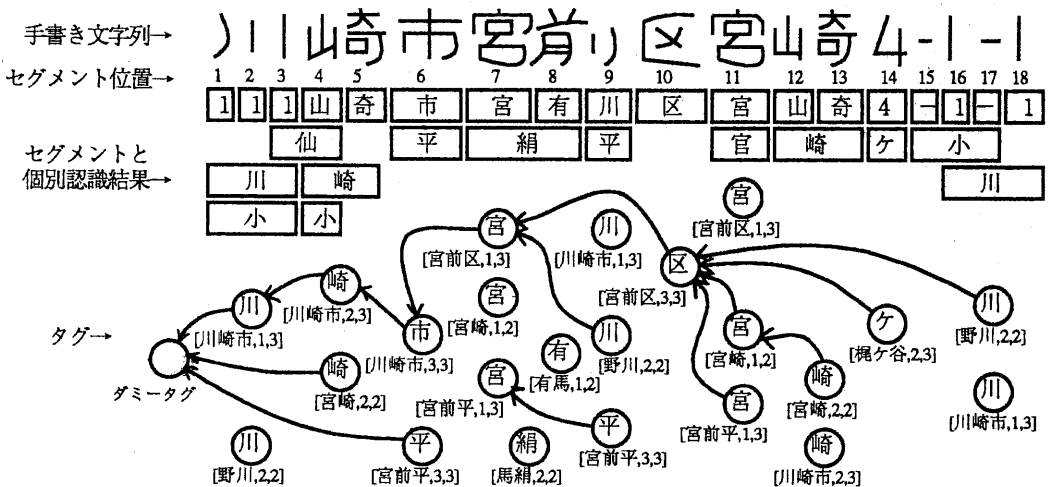


図3 文字タグ法の処理イメージ

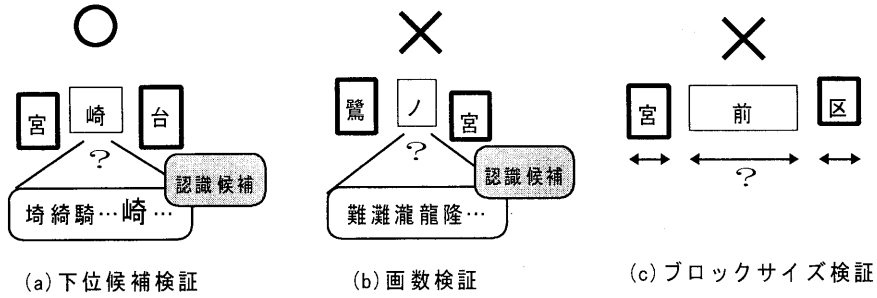


図4 多様な簡易検証方式

しかし、手書き文字列読み取りにおいて再処理が有効でない場合もある。なぜなら、パタンの変形や接触が激しく、どうしても切り出せない／認識できない文字の発生が避けられないからである。このような場合、人間は文字を厳密に読んでいるのではなく、前後の文字から推測される文字を該当パターンと対比し、極端に異常と思わなければその解釈を受け入れていると考えるのが妥当であろう。そのような枠組みを用意することで、読み取り性能の向上が期待できる。

また別の観点として、処理時間の問題がある。一般にボタン処理は処理量が大きく、すべての検証必要箇所に対して再ボタン処理を施すと、膨大な処理時間となるおそれがある。

そこで、検証系の構成は真の再ボタン処理以外にいくつかの簡易的な検証を用意し、それと組み合わせる方式を提案する。例えば、簡易的な検証として、次のようなものが考えられる。

下位候補検証：文字認識候補の下位に知識から推測される文字が存在するかどうかを検証する（図4(a)）。

画数検証：認識不良領域の候補文字の画数が知識から期待される文字の画数に見合うかどうかを検証する。ストロークが1本しかないような文字と非常に複雑な文字とを区別することができる（図4(b)）。

ブロックサイズ検証：認識不良領域のサイズが知識から期待される文字数に見合うかどうかを検証する。認識不良の前後いくつかの文字のサイズから期待されるブロックサイズ

を計算し、それと比較する（図4(c)）。

4. 手書き住所読み取りへの適用

提案したB/T融合型アプローチに基づく手書き住所読み取り実験システムを試作し、主に地名部分の読み取り精度の評価を行った。なお、実験システムは実用的場面も意識し、単に処理結果の最尤候補を出力するという方式ではなく、読めない画像をリジェクトする機能も採り入れた。

4.1 実験システムの構成

実験システムの全体構成を図5に示す。ボタン処理結果を入力として知識処理は動作する。地名部は文字タグ法（2節参照）をベースに候補抽出を行い、必要に応じてトップダウン的な検証処理を起動する。本システムでは、街区部の知識処理ルーチン[8]も結合しており、読み取り結果として街区までを出力することが可能である。街区部は地名部からサブルーチンのように起動され、町名候補ごとの街区の値の範囲を制約条件として受け取り、街区として認定された文字列を出力する。

4.2 候補選択処理

地名の候補選択は次の手順で行う。

[Step1]文字候補中から、文字タグ法で評価値の高い地名を5候補までを抽出する。文字認識上位5候補までを使う。

[step2]最下層の地名要素（通常は町名）の各文字に対して次のような得点積算を行う[9]。

- ・文字が読めた場合 +W1
- ・文字が読めなかった場合 -W2

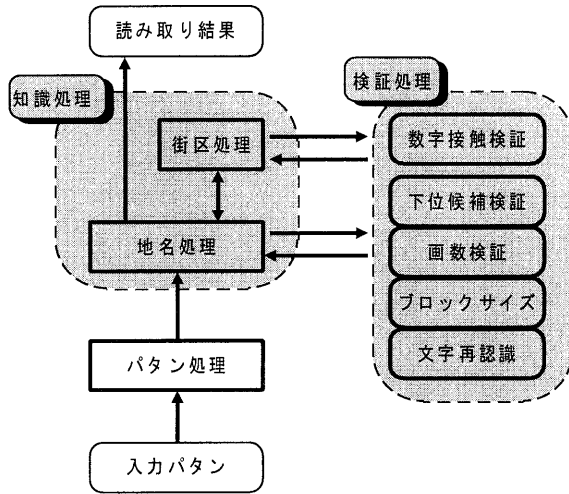


図5 実験システムの構成

得点計算の確信度を高めるために、前後の住所要素が正しく読めたという状況も、町名の得点積算に加えている[9]。具体的には、町名の上位要素（通常は市区名）、あるいは下位要素（街区）が読めた場合、町名が1文字読めたのと同等の加点をする。

[step3]読み取り文字得点にもとづいて最大5候補の住所候補を挙げた後、各候補についてパターン処理がうまくいかずに知識から推測した箇所のトップダウン検証を行う。この結果も読み取り文字得点に反映させ、最終的に最高得点を獲得した町名（上位の住所文字列を含む）を判定結果として出力する。読み取り状況得点に対しては町名ごとにしきい値を設定しておき、それを超える候補が一つもない場合はリジェクトとする。また、同点の候補が複数ある場合もリジェクトとする。

4.3 トップダウン処理

トップダウン処理としては、次のものを用意した。

再切り出し／認識検証（アラビア数字接触検証）：知識から推測できる文字に該当するブロックを再度文字認識する。複数文字が1ブロックに該当するときは領域分割も行うが、現在は2文字までしか想定していない。結果として正しく読めれば、該当文字の得点を、不一致点数-W2 から+W1 に変える。また、

この処理は地名部だけでなく街区部にも導入しており、アラビア数字接触と推測できるブロックがある場合、2文字の接触と仮定して強制分離／再認識する[6]。この結果として街区が正しく読み取れると、町名の下位要素が読めたとして、得点+W1 を加算することになる。なお、この処理は次に述べる簡易的な検証にくらべて処理時間が多くかかるので、他の検証を行っても判断がつかない場合にのみ起動する構成とした。

下位候補検証：知識から推測できる文字が、文字認識下位10候補までに存在すれば、該当文字の得点を不一致点数-W2 から+W1 にする。

画数検証：知識から推測できる文字に対し、該当ブロックの文字認識候補の平均画数が大きく異なれば、該当文字の得点を-W2 から-W3 に変える。W3 は検証におけるペナルティ得点である。実験では「どちらかの画数が4画以下で、かつ差が5画以上ある場合」をペナルティ付与条件とした。

ブロックサイズ検証：知識から推測できる文字数の該当するブロックサイズが、その前後ブロックのサイズと比較して著しく不適切と判断した場合、不一致得点-W2 を-W3 に変える。W3 は検証におけるペナルティ得点である。現在は、前後ブロックの平均幅の2倍以上のブロック、あるいは1/3以下のブロックにペナルティを付けている。

4.4 実験および評価

評価は地名（町名）の読み取りに関して、(1)文字タグ法単独の1位正解率、(2)トップダウンなしの読み取り性能、(3)トップダウンありの読み取り性能、を比較する形で行った。対象とした画像は葉書の縦書き住所で、東京都国分寺市（セットK）と静岡県富士市（セットF）宛、各300画像である。町名数はそれぞれ24と205である。パターン処理においては、文字領域の切り出しに石寺らの方式[10]を、文字認識に津雲の方式[11]を用いた。また、トップダウン検証での文字の再認識には、浜中の方式を用いた[12]。町名ご

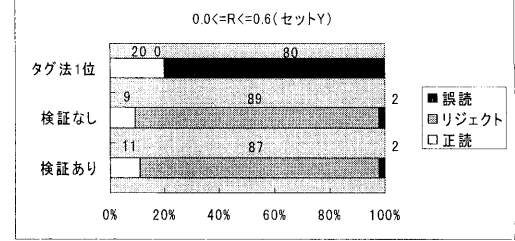
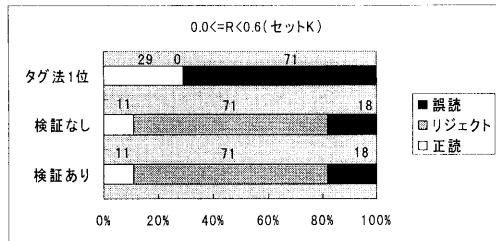
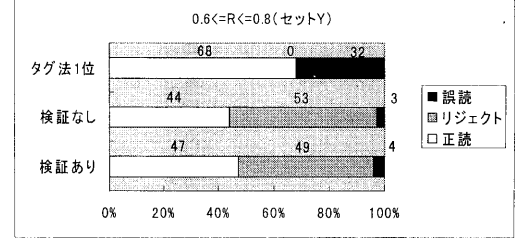
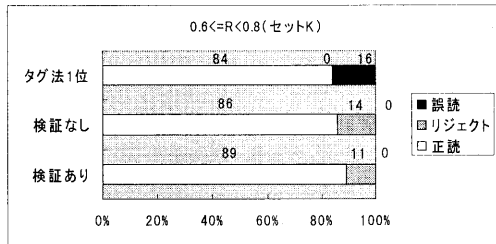
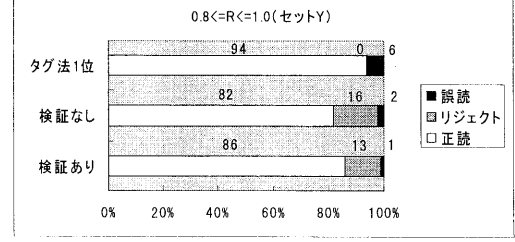
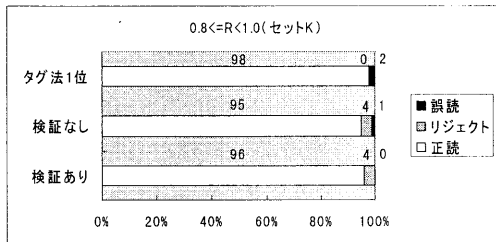
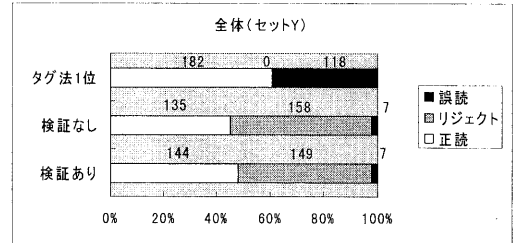
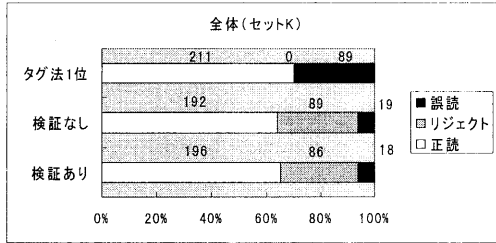


図 6a 実験結果 (セット K)

図 6b 実験結果 (セット Y)

とのリジェクトしきい値は、経験的に、2文字町名するとき、 $W1+2$ 、3文字町名するとき $W1+3$ 、4文字町名するとき $W1+4$ とし、 $W1$ から $W3$ はそれぞれ、 $W1=3$ 、 $W2=1$ 、 $W3=3$ と設定した。実験結果は、各入力パターンをパターン処理結果の正解文字含有率 R によって3種類 ($0 \leq R < 0.6$ 、 $0.6 \leq R < 0.8$ 、 $0.8 \leq R \leq 1.0$) に分類してまとめた。知識処理への入力の状態による性能の違いを見るためである。

実験結果を図 6a (セット K)、図 6b (セット Y) に示す。それぞれ、上からセット全体、 $0.8 \leq R \leq 1.0$ 、 $0.6 \leq R < 0.8$ 、 $0.0 \leq R < 0.6$ の

性能となっている。

4.5 定量的考察

(a) 文字タグ法の候補抽出能力

正解含有率 R が 0.8 以上では、文字タグ法により 1 位候補に 94%、98% という高い割合で正解候補が出力されている。また、 R が 0.6 ~ 0.8 という比較的低い場合も、68%、84% という高率であり、候補文字の抜けがあっても正しく候補を挙げることができる文字タグ法の特長がよく表れた結果となった。文字タグ法自身に棄却機能を入れていないため、

R が低いと誤りが多くなるが、地名しきい値によるリジェクト処理で対処可能である。

(b) トップダウン検証の効果

トップダウン検証による性能改善は、セット K 全体で正読+4 (192 → 196) かつ誤読-1 (19 → 18)，セット Y では、正読+9 (135 → 144) かつ誤読増減なし(7 → 7) であった。全体で見ると量的効果は小さいが、R が 0.6 以上、すなわち中程度以上のところによく利用している。定量的効果の小ささは、しきい値によるリジェクト能力が強すぎたことも原因である。

効果の内訳で目立ったのは、リジェクトから正読に変化したもの (10 件) であった。同点競合によるリジェクトやしきい値ぎりぎりでのリジェクトが、下位候補検証で救われた例が多かった。一方で、リジェクトを誤読とする改悪も合計で 4 通ほど見られた。

実際に性能改善に効果のあった検証の多くは下位候補検証、画数検証、ブロックサイズ検証によるものである。真の再パタン処理が起動されても、ほとんどが判別不能という結果になってしまった^{*1}。手書きに対し知識情報からトップダウン検証を行う場合、たとえ簡易な検証処理であっても適切に起動すれば、十分な効果のあることが実証されたといえる。

(c) 処理時間

処理時間を NEC 製 EWS4800/330 ワークステーションで測定した。その結果、文字タグ法による探索はすべて 1 住所あたり 20ms 以内 (タイマの精度が 10ms) であり、ボトムアップ処理の速度として十分と思われる。また、検証に関しても文字の再認識は数十 ms (ソフトウェア) がかかるが、下位候補検証、画数検証、ブロックサイズ検証は非常に軽い処理であり、文字タグ法と比べて問題に

ならない処理時間であると考えられる。

4.6 検証処理効果の定性的分析

次に、改善と改悪の実例 (図 7, 8) を示して議論する。図の読み取り候補文字列に付いている「○」は、文字が個別文字認識候補中に存在したか検証に成功したもの (得点+W1)，「△」は個別文字認識候補中に存在しなかったもの (得点-W2)，「×」は画数検証やブロックサイズ検証によって文字がリジェクトされたもの (得点-W3) を意味する。

(a) 下位候補検証の成功例 (図 7)

検証を用いない場合、「日吉」という町名と「内藤」という町名が、それぞれ(7,1)，(8,1)で 2 文字中の 1 文字ずつ認識候補にあり、同点競合となる。トップダウン検証を行うと、下位候補検証によりブロック(7,1)の下位候補に「内」が見つかり、正しく「内藤」と判定できる。

(b) 画数検証による改悪例 (図 8)

画数検証により、競合リジェクトが誤読になった例である。ブロック(7,1)「東」が略字で書かれているために文字認識不良となる。「東元町」と「西元町」が候補に挙がっても、

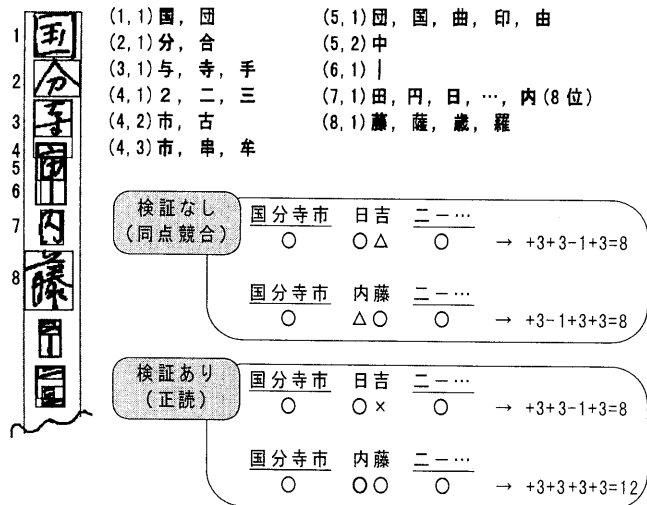


図 7 検証による改善例

*1 速度の問題から簡易検証を先に実行したため、結果的に真の検証が起動された例が少なく、効果が現れなかったという側面もある。

検証がなければ同点で競合する。これに対し検証を行うと、(7,1)のブロックに対する認識候補の画数が小さいため、「東」が画数検証によりペナルティを受け、結果として「西元町」に誤読してしまう。

この例に限っては、表記が「東」の略字であるため、画数が少なくなったことが原因であると言える。しかし、画数検証は文字領域の「黒さ」を画数で近似しているため、その近似能力に限界がある。精度向上のためには、検証のバリエーションを増やすことも重要な課題となる。

5. おわりに

本稿では、ボトムアップ／トップダウン融合型の知識処理方式について述べ、それを住所読み取りに適用した際の評価を述べた。パターン処理の多数候補の組合せを効率よく探索する文字タグ法、処理量の少ない検証処理を中心とした検証系の実現によって、住所読み取りに対して高速かつ詳細な知識処理を行えることを実証するとともに、提案方式の有効性を示すことができた。

今後は、ボトムアップ検証処理適用条件の詳細化、ボトムアップ検証処理のバリエーションの開発等が課題となる。

謝辞

本研究を進めるにあたって、パターン処理（文字列領域検出、文字切り出し、文字認識）の技術を検討して下さった、山田課長をはじめとする NEC 情報メディア研究所パターン認識研究部各位に感謝します。また、実験用画像や地名データを用意して下さるとともに、有益なコメントをいただいた NEC 産業オートメーション事業部、NEC ポスタルテクノレクス（株）の各位に感謝します。

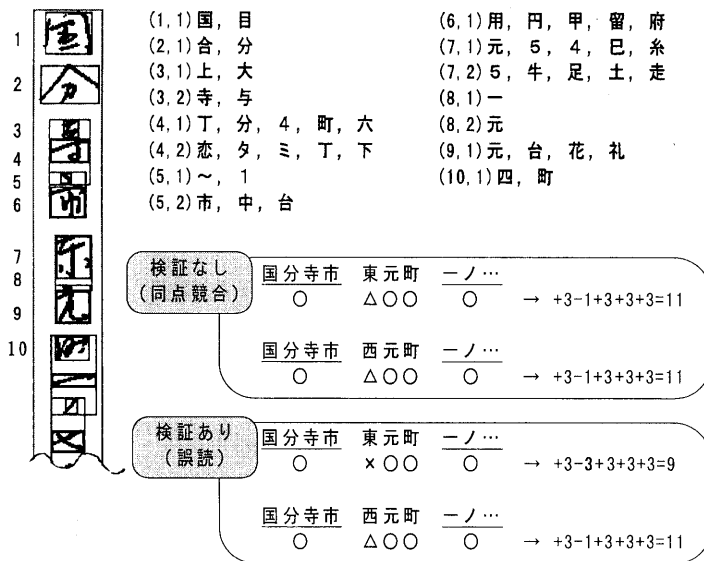


図8 検証による改悪例

参考文献

[1]高尾他:日本語文書リーダ後処理の実現と評価, 情処論文, Vol30, No11, 1989
 [2]伊藤他:OCR 入力された日本語文の誤り検出と自動訂正, 情処論文, Vol33, No5, 1992
 [3]木谷:手書き文書の文字認識結果に対する後処理方式, 情処研究会, 91-NL-86, 1991
 [4]佐瀬他:制限付き文字列読み取りの一検討, 信学技報, PRU88-115, 1988
 [5]仲林他: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 回全国大会, 4E-8, 1987
 [6]下村他:手書き住所読み取りにおけるパターン処理と連携した住所知識処理方式, 情処第 50 全国大会, 4D-1, 1994
 [7]福島他: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—, 情処論文, Vol37, No4, 1996
 [8]下村:手書き住所読み取りにおける街区住所知識処理方式, 第 51 情処全大, 4R-8, 1995
 [9]山内他:多段弾性照合法の郵便あて名区分機への応用, 信学システムソサエティ大会, D-200, 1995
 [10]石寺他: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
 [11]津雲:手書き漢字認識, NEC 技報, 47-8, 1994
 [12]濱中他:変形推定を用いた検証型文字認識の検討, 信学総大, D-542, 1995