仮説生成と検証の効率的組合せに基づく手書き文字列読み取り向け知識処理方式

下村秀樹[†] 福島俊一[†] 山内俊史[††]

本論文では，個別文字の切り出し/認識の候補からボトムアップ処理で読み取りの文字列候補（仮説）を抽出する「仮説生成」処理と，読み取り対象に関する知識からのトップダウン的な「検証」処理との効率的な組合せに基づく，手書き文字列読み取り向けの知識処理方式を提案する．従来，文字列読み取りの知識処理は，個別文字の切り出し/認識の候補をボトムアップに知識と照合し，読み取り結果を出力する方式が主流であった．しかし，単なるボトムアップ処理だけでは，限られた処理時間内で読み取り精度を高めることに限界がある．本論文では，これに対して，尤度の高い個別文字切り出し/認識候補からの仮説生成と，仮説中の不確実な部分だけを詳細に確認する検証との適切な組合せが，短い処理時間で高い読み取り精度を達成しうるアプローチであると主張する．また，その要素技術として，文字タグ法による効率の良い仮説生成と，下位候補検証，画数検証，領域サイズ検証という3種類の簡易検証方式を提案し，それらを組み合わせた住所読み取りシステムを試作した．知識処理フェーズに閉じた評価ではあるが，仮説生成に使用する文字認識上位候補数を変えながら，知識処理時間と読み取り精度を測定する実験により，仮説生成と検証の組合せが，単なるボトムアップよりも短い処理時間で高い精度を達成しうることを確認した．

A Knowledge-based Processing Method that Uses an Efficient Hypothesizing and Verification Hybrid Approach for Handwritten Character String Reading

HIDEKI SHIMOMURA,[†] TOSHIKAZU FUKUSHIMA[†] and TOSHIFUMI YAMAUCHI[††]

This paper proposes a knowledge-based processing method based on a hypothesizing and verification hybrid approach for reading handwritten character string. Most of the conventional methods are based on the bottom-up approach, which simply searches through pattern processing results for word sequences matching with the knowledge. However, it is difficult to improve the reading accuracy on this approach, with limitations of processing time. For this problem, this paper insists that an appropriate combination of hypothesizing and verification processes enables the higher reading accuracy in a shorter time. It also proposes a efficient hypothesizing method and three simple methods for the verification. They are applied to a handwritten address reading system. Experimental results show effectiveness of the proposed approach, at least in the knowledge processing phase.

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個別文字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いて個別文字切り出し/認識の結果を補完する「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（以下，単に「枠なし」と略す）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こりやすく，個別文字切り出し/認識（以下，本論文では「パタン処理」☆☆と呼ぶ）が非常に難しいた

† 日本電気株式会社ヒューマンメディア研究所
Human Media Research Laboratories, NEC Corporation
†† 日本電気株式会社産業オートメーション事業部
Industrial Automation Division, NEC Corporation

☆「知識処理」が扱う「知識」には，読み取り対象の文字列を構成する単語の辞書，単語間の接続規則などが考えられる．また，やや特殊ではあるが，住所読み取りにおける各町名に付帯する丁目・番地の値の範囲も知識と考えられる．

☆☆「知識処理」に対し「パタン処理」は，画像的特徴に基づく処理であること，文字列全体ではなく局所的情報による処理であることを主な特徴とする．

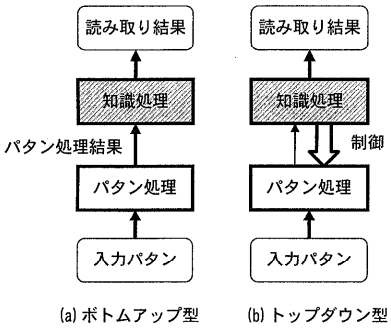


図1 従来の代表的な文字列読み取り知識処理方式
Fig. 1 Conventional knowledge-based processing ways for character string reading.

め,「知識処理」に対する期待は大きい．なお，以下本論文において,「(個別文字) 認識」は個別文字の識別,「(文字列) 読み取り」は記載された文字列の判定を意味するものとする．また文字列読み取り処理によって出力される文字列を,「読み取り文字列」と呼ぶ．

従来の文字列読み取り知識処理は，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)~6)]．すなわち，パタン処理が出力する候補群と，文字列を構成する単語の辞書や単語間接続関係といった知識とを照合し，最もよく合致するものを結果とする直列型の構造である．この方式は主に，パタン処理が比較的容易な印刷や記入枠制約ありの文字列読み取りを対象として検討が進められてきた．しかし，一般に，枠なし手書き文字列を対象にするとパタン処理の精度が低くなることから，パタン処理で多くの候補を出力しないと高い正解含有率☆を実現できない．ボトムアップ型ではそれにともなって知識処理の時間が単純に増加してしまうので，限られた処理時間の中で高い精度を出すことに限界がある．

ボトムアップ型の拡張として，まず読み取り文字列の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[7)~9)]．しかし，処理を単にやり直すだけでは，効果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も難しい．

一方，読み取り対象を非常に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ン処理を制御する方式も提案されている[10)]．しかし，トップダウンの根拠となる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も高いため，一般にはあまり用いられない．

これらトップダウン処理とボトムアップ処理の欠点を補うため，両者を融合させる発想も生まれており[11)]，筆者らも枠なし手書き文字列読み取りを対象としこれに取り組んでいる[12)]．本論文では，ボトムアップ処理を読み取り文字列の仮説生成，トップダウン処理をその仮説の検証ととらえ，この処理方式を「仮説検証型アプローチ」と呼ぶ．

仮説検証型アプローチをボトムアップ処理の後に検証処理を追加したものと見るならば，検証処理の効果の分だけボトムアップ処理よりも高い読み取り精度が期待できる．しかし，ボトムアップ型でも，最初から検証処理で使うのと同レベルの情報量/候補数を用いるなら，同程度の読み取り精度が達成できるのではないか，と考えてもおかしくない．こう考えてみると，仮説検証型アプローチの優位性は，限られた処理時間の中で高い精度を達成できるという，処理時間と読み取り精度のバランスにあると考えるのが妥当である．すなわち，高い読み取り精度を達成しようとした場合，ボトムアップ型アプローチではすべての箇所で詳細な解析を実行することになるのに対して，仮説検証型アプローチでは必要な箇所でのみ詳細な解析を実行すればよいことになる．

しかしこれまで，処理時間と読み取り精度の観点から，仮説検証型のアプローチを議論あるいは評価した例は見当らない．したがって，仮説生成や検証に必要な要素技術，仮説生成と検証の組み合わせ方を検討し，それらが処理時間と処理精度に与える効果を示すことは，文字列読み取り知識処理に新たな視点を与えるものとして重要な意味がある．

以上の前提をふまえ本論文では，仮説検証型アプローチが，短い処理時間で高い精度を実現する可能性を持つ手法であることを明らかにするとともに，そのための要素技術を検討・提案し，さらに，処理時間と精度の関係からその利点を裏付ける実験についても報告する．まず，2章では，仮説検証型アプローチが，短時間で高い読み取り精度を実現する可能性を持つ手法であることを，従来技術と対比しながら議論し，明らかにする．次に，3章では，仮説検証型アプローチを構成する要素技術として，仮説生成と検証の具体的なアルゴリズムを例示する．ここで提案する検証のアルゴリズムは，パタン処理の再起動までは求めないタイプのものなので，試作した手書き住所読み取りシステム上で行った実験（4章）では，知識処理フェーズの処理時間と読み取り精度のバランスを評価することで，仮説検証型アプローチおよび要素技術の効果を考

☆ 読み取り対象の全文字数に対して，パタン処理結果の文字候補中に正解が含まれている文字の割合．文字候補の総数が増えるほど正解含有率が高くなる傾向を示す．

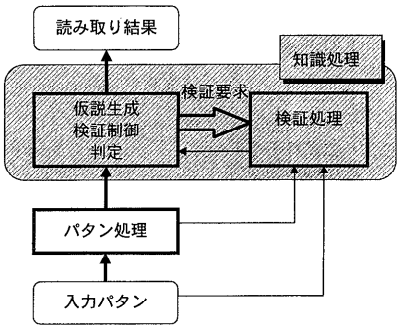


図2 仮説検証型アプローチ
Fig. 2 The hypothesizing and verification hybrid approach.

察する．

2. 仮説検証型アプローチ

2.1 処理構成

仮説検証型アプローチに従った文字列読み取り処理の構成を図**2**に示す．図1に示した従来技術のボトムアップ処理を読み取り文字列仮説の生成処理と見なし，それにトップダウンの検証処理を加えた構成になっている．検証処理を起動する位置は，知識から見てパタン処理が不完全と思える箇所である．たとえば，知識と一致する文字認識候補が存在せず，文字の切り出しや認識が正しく行われていないと判断できる箇所などが，検証処理の対象となる．検証要求では，検証の対象位置と知識からその位置に推測できる文字列を指定して，検証処理を起動する．検証処理の結果としては，推測文字列が存在すると見なせるか否かの判定結果，あるいは存在の可能性を示す値などを返す．知識処理内部では，その情報を加味して仮説の採否や競合する仮説間の優劣判定を行い，最終的な読み取り文字列を決定する．

1章においてボトムアップ型の拡張としてフィードバックループ（処理のやり直し）を導入した方式があることに触れたが，これと仮説検証型の間では再処理時の考え方に大きな違いがある．従来型のフィードバックでは，知識に合致しない箇所への再処理において知識情報の積極的な利用が見られない．これに対し，仮説検証型アプローチでは，知識からの推測をトップダウン的に与える．すなわち，「この領域にはこの文字が書かれているのではないか」という観点に基づいた処理を行える点に特長がある．

2.2 処理時間と読み取り精度のバランス

1章で述べたように，本論文では処理時間と読み取り精度のバランスという観点から仮説検証型アプローチの優位性を論じる．そのために本節では，文字列読み取りにおける各過程が入力とする候補数/情報量に着目して，ボトムアップ型と仮説検証型を比較する．

ボトムアップ型による文字列読み取りでは，文字切り出し処理→文字認識処理→知識処理という処理の流れになる．前半の文字切り出しと文字認識がパタン処理にあたる．これらの各過程の処理時間は，各過程（文字切り出し/文字認識/知識処理）が入力とする候補数/情報量に大きく左右される．通常，扱う候補数/情報量が増えるほど，各過程の処理時間は増加する．**図3**にその概略イメージを示す☆．たとえば，文字認識過程について，文字切り出し結果の上位候補を対象とした文字認識処理（R1）を基準として考えると，文字切り出し結果の下位候補を対象とした文字認識処理（R2）まで行うなら文字認識処理時間はそれだけ増加する．さらに，R1・R2とは異なる文字認識方式を用いた文字認識処理（R3）まで行う（複数通りの文字認識方式を適用してみる）ならば，文字認識処理時間はいっそう増加する．

さて，ボトムアップ型による S1 × R1 × K1 という処理の流れを基準に考えてみる．すなわち，与えられた領域画像を対象とした文字切り出し処理（S1）を行い，その文字切り出し結果の上位候補を対象とした文字認識処理（R1）を行い，その文字認識結果の上位候補を対象とした知識処理（K1）を行うという流れである．このようなボトムアップ型での読み取り精度を高めようとすると，前段の処理から後段の処理へなるべく多数の候補/情報を送り出すようにしてやる必要が生じる．その結果，図3の S2·R2·R3·K2·K3 など処理をふくらませることになり，全体の処理時間は (S1 + S2) × (R1 + R2 + R3) × (K1 + K2 + K3) のような組合せに応じて増大することになる☆☆．

これとの比較として，S1 × R1 × K1 というボトムアップ処理を仮説生成に用いた仮説検証型を考えてみる．検証処理をどのように実現するかについては後述する（3.2節）が，S1 × R1 × K1 による仮説生成に対しては，S2 · R2 · R3 · K2 · K3 などによる検証が

☆ 図における S1 と S2，R1 と R2 と R3，K1 と K2 と K3 の間の区分（上位候補と下位候補をどこで区切るか，アルゴリズム/パラメータのどの程度の違いで異なる方式だと区別するかなど）について，厳密な線引きはせずに議論することを容赦願いたい．ここでは，扱う候補数/情報量（処理の詳細度）を粗くいくつかの段階に分けることで，概略イメージをつかみやすくしたものである．

☆☆ ここでは，全体の処理時間が厳密にこの式によって計算できると主張しているわけではない．この式は処理の組合せを表しており，このような処理の組合せが大きな要因となって，全体の処理時間の増大をまねくだろうと考えている．

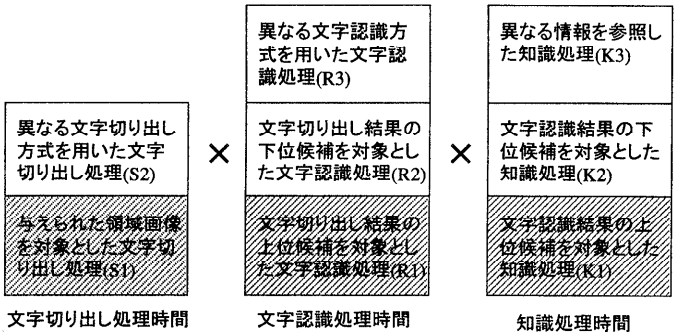


図3 文字列読み取りにおける各過程の詳細度と処理時間の関係
Fig. 3 Relation between completeness and time of reading process.

考えられる．検証処理は知識から見てパタン処理が不完全と思える箇所に限定して起動するので，a，b，c，d，eを局所的な実行を意味する係数（1より小さい正の数）とすれば，全体の処理時間は $(S1 + a \cdot S2) \times (R1 + b \cdot R2 + c \cdot R3) \times (K1 + d \cdot K2 + e \cdot K3)$ で済むことになる．したがって，同レベルの候補数/情報量を用いているなら同程度の読み取り精度を達成しうると仮定すると☆，仮説検証型は，同程度の精度を達成するのに，ボトムアップ型よりも短い処理時間しか要しないであろう．別ないい方をすると，仮説検証型によれば，限られた処理時間を，詳細な解析を必要とする箇所に適切に配分できるため，短い処理時間で高い読み取り精度を実現しうる．

3. 仮説生成と検証のアルゴリズム

3.1 仮説生成：その条件とアルゴリズムの例

読み取り仮説の生成方式に求められる要件として，仮説のもれをなくすために，(1) パタン処理結果の複数候補を扱えること，(2) 一部正解文字の欠落を許容できること，また一般的要請として，(3) 高速であること，があげられる．これに対して筆者らは，「文字タグ法」をすでに提案している[13]．ここでは，「文字タグ法」の概要とそれが上記の要件を満たすことを簡単に説明する☆☆．

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文字認識候補に対して，文字列を構成しうる単語を格納した単語辞書を参照し，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： 末端のタグから接続をたどり，尤度の高いいくつかの候補を出力する．

図4は，文字タグ法を住所の市名～町名部分の読み取りに適用した場合の処理イメージである．これに従って，文字タグ法の動作をさらに説明する．

まず，(a) の手書き文字列に対して個別文字切り出しと個別文字認識を行い，それぞれ複数の認識候補を得る．(b) のセグメント位置は，切り出された文字の位置を，切り出し最小単位（セグメント）の番号によって示したものである．セグメント番号は，文字切り出し処理によって，文字列記載方向に向かって1から順に振っている．この番号の若い方が，文字として位置的に前にあることを意味する．連続する複数セグメントで構成される切り出し候補もある．(c) の長方形で囲まれた文字が切り出し領域に対する1つの認識候補文字である．たとえば，記載文字列の先頭の「川」という文字に対しては3つに分割された「111」，あるいは正しい切り出しに対する「小」「川」などが候補として出ている．これが文字タグ法による処理への入力となる．

まずstep1では，得られたすべての候補文字について，単語辞書を参照しながら，どんな単語の何文字目になるかというタグを生成する．この住所読み取りの例では，市名，区名，町名がそれぞれ単語に該当する．図中の丸で囲まれた認識候補文字に対して，直下の括弧で囲まれた3つ組が生成されたタグを示している．

☆ 処理時間に関する比較は無視して，同レベルの候補数/情報量を用いたときに，ボトムアップ型と仮説検証型とでは，どちらの方が高い読み取り精度を達成しうるアプローチであるかは，慎重な検討と評価を要する研究課題である（5章において，今後の課題としてあげ，その理由を述べる）．

☆☆ もちろん，仮説生成には別の手法を用いてもかまわない．ここでは，仮説生成の要求を満たす技術が実際に存在することを示すとともに，4章の実験の理解を助けるため，文字タグ法を説明している．

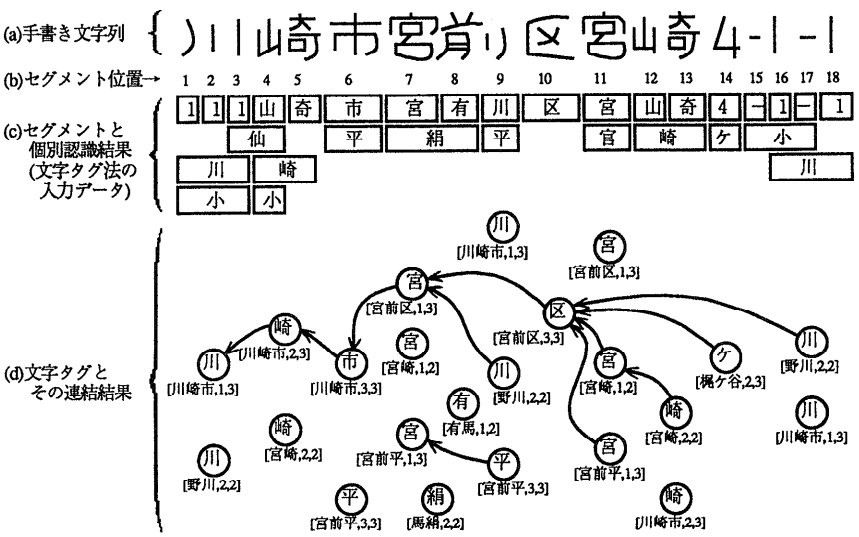


図 4 文字タグ法の処理例
Fig. 4 An example of the word sequence search process.

先ほどの「川」に対しては，[川崎市,1,3] あるいは [野川,2,2] というタグが生成されており，それぞれ「『川崎市』の 3 文字中の 1 文字目」,「『野川』の 2 文字中の 2 文字目」であることを意味する．同じ文字領域に対して「小」も候補に出ているが，この文字を含む単語が辞書に存在しないので，タグは生成されない．

次の step2 では，前側のタグから順にすべてのタグについて，位置的にその前方に存在するタグの中で最尤のものを接続する処理を繰り返し，読み取り候補の単語列（以下,「候補単語列」）を作成する．この際，単語列を構成する一部文字の欠落によってタグが位置的に不連続であっても，接続可能性を検査する．この文字の欠落の有無は，候補単語列の尤度（もっともらしさを表す数値）に反映させる．たとえば，記載文字列の 6 文字目の「区」という文字に対応するタグ [宮前区,3,3] に関しては，[川崎市,1,3]，[野川,2,2]，[宮前区,1,3]，[宮崎,1,2] をはじめとして，多数のタグがその前方に存在する．それらすべてについて，[宮前区,3,3] と接続した場合の尤度を計算し，最尤のものに接続する☆．尤度は同一単語内，あるいは連鎖可能な単語間の文字のタグを，できるだけ飛ばさず位置的に連続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実際に飛ばした文字領域の数が一致しているほど尤度を高くする．このタグの飛ばしが，柔軟な不完全一致を可能とする．

この例では，5 文字目の「前」という文字が正しく認識されていないが，読み飛ばした文字数とその間にできた領域数がそれほど異ならないため，[宮前区,3,3] から [宮前区,1,3] への接続が最良となった．

最後の step3 では，どのタグからも接続されていない末尾のタグを尤度の順にソートし，上位いくつかの候補を出力する．さきに述べた尤度設定の考え方から，尤度は一般的に，より多くのタグを接続した単語列ほど高くなる．言い換えれば，パタン処理出力中の文字候補と，より多く一致した単語列候補の尤度が高くなる．この例では,「川崎市宮前区宮崎」が 1 位候補となる．

このように文字タグ法では，単語の構成情報をタグの中に巧妙に組み込んで，文字切り出しの曖昧性と個別文字認識の曖昧性を同時に扱いながら，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とを接続していくことによって，最尤単語列を効率良く探索できる枠組みを提供している☆☆．パタン処理が出力する大量の候補，正解文字の欠落に対しても効率的に

☆ 非常に離れたタグを接続すると尤度が低くなるので，現実には近傍のタグだけを見れば十分である．

☆☆ 従来の代表的なボトムアップ処理方式は，まず文字認識候補の組合せに対して単語辞書を照合し，得られた単語候補の組合せをビタビアルゴリズム（動的計画法）を用いて探索するという 2 段階構成である．正解文字の欠落に対処するためには，単語辞書との照合の段階で 1 文字欠落・2 文字欠落などの様々な虫食いパタンを調べなくてはならなくなる．さらに，文字切り出しの曖昧性を個別文字認識の曖昧性と切り離して扱うものが多く，文字切り出しの曖昧性の分だけ反復処理が必要となる．文字タグ法は実用上，入力文字列長に対して線形の計算量で実行可能であるが，従来法でこれと同じ広い探索空間を扱おうとすると，計算量が組合せ的な爆発を起こす[13]．

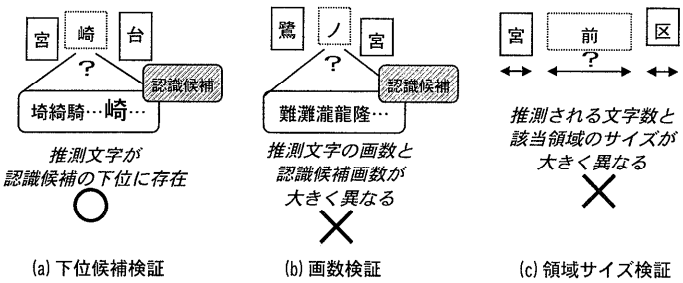


図5 簡易検証方式
Fig. 5 Simple verification methods.

対処できることから，前述の仮説生成アルゴリズムに求められる要件を満たしている．

3.2 検証：簡易な検証方式の具体例

2.2節で説明した図3に従えば，検証処理のタイプとして，異なる文字切り出し方式を用いた文字切り出し処理（S2），文字切り出し結果の下位候補を対象とした文字認識処理（R2），異なる文字認識方式を用いた文字認識処理（R3），文字認識結果の下位候補を対象とした知識処理（K2），異なる情報を参照した知識処理（K3）などが考えられる．このうち，本論文では，K2とK3のタイプの検証処理を実現する方針とした．

K2・K3とも知識処理のフェーズで実行する検証であるから，次節以降では，知識処理フェーズに要する時間と読み取り精度とのバランスを議論していく．2.2節で述べたように，仮説検証型アプローチは本来，パタン処理から知識処理までの全体処理時間に着目して，精度との良いバランスを実現できる枠組みである．次節の実験は，その有効性を，まず知識処理フェーズのファクタに絞った形で実証を試みたという位置付けになる．

また，まずK2やK3のタイプに該当する比較的簡易な検証を適用し，そのような簡易検証では判定できなかった箇所に絞って，R2やR3のタイプ，さらにはS2のタイプを適用するという多段階の検証戦略をとれば，限られた処理時間をより有効に活用できるものと考える（K2・K3タイプに比較して処理量の大きいR2・R3・S2タイプの検証処理の適用箇所を効果的に絞り込める）．

K2・K3タイプの検証処理として，具体的には，以下に示す下位候補検証，画数検証，領域サイズ検証を考案した（**図5**参照）．K2タイプ（文字認識結果の下位候補を対象とした知識処理）に該当するのが下位候補検証であり，K3タイプ（異なる情報を参照した知識処理）に該当するのが画数検証と領域サイズ検証である．

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5(a)）．パタン処理では多くの文字候補を出すが，知識処理ではその上位だけを使って仮説を生成する場合の検証として，効果が期待できる．

画数検証： 知識から期待される文字の持つ画数が，それに該当する文字切り出し領域の文字認識候補の画数と見合うかどうかを検証する．たとえば，知識から期待される文字の画数と，該当文字領域の文字認識候補の平均画数とを比較し，一定以上差があれば仮説を棄却するなどが考えられる．これによって，ストロークが1本しかないような文字を非常に複雑な文字に誤って解釈するようなケースを排除できる（図5(b)）．

領域サイズ検証： 知識から期待される文字の数に，対応する文字切り出し領域のサイズが見合うかどうかを検証する．具体的には，検証対象の文字領域の前後いくつかの文字領域サイズから，その領域に何文字が書かれていそうかを推定し，それが知識から期待される文字数と大きく異なれば仮説を棄却する（図5(c)）．

これらの検証処理は，パタン処理の再起動までは行わないので処理量が比較的少ないということに加えて，仮説を採択すべきかどうかを判定する視点だけでなく，棄却すべきかどうかを判断する視点も導入していることも特長である[12]．また，人間が癖字やつづけ字に対して判読が難しいとき，必ずしも文字を厳密に読もうとはせずに，知識から推測される文字を該当パタンと対比し，極端に異常と思わなければその解釈を受け入れるようなケースにも似ている．

4. 手書き住所読み取りを対象とした評価実験

3章で述べた要素技術を実装した仮説検証型アプローチに基づく手書き住所読み取りシステムを試作し，地

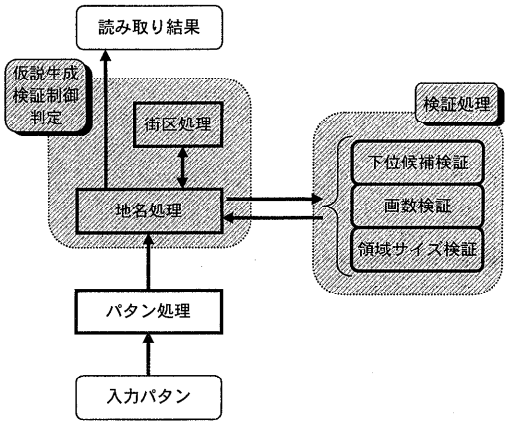


図6 実験システムの構成
Fig. 6 The structure of the experimental system.

名☆の知識処理時間と読み取り精度の評価を行った．

4.1 実験システムの構成

実験システムの全体構成を図6に示す．パタン処理結果を入力として知識処理は動作する．地名処理では，文字タグ法（3.1節参照）により候補抽出を行い，それを読み取り文字列の仮説とし，必要に応じて検証処理を起動する．本システムでは，街区☆☆の知識処理ルーチン[14]も結合している．街区処理は地名処理で街区を持つ地名単語が候補にあがった場合にサブルーチンのように起動される．そして，街区の値の範囲を制約条件として受け取り，街区に関する知識，制約条件に合致した文字列を出力する．街区の処理結果は，それに接続する地名単語の尤度計算に反映させる．

4.2 候補選択処理

地名の候補選択は次の手順で行う．

Step1：仮説生成 パタン処理結果の候補空間を文字タグ法で探索し，文字タグ法での尤度の高い地名単語列を5候補まで抽出する．これを，読み取り文字列の仮説とする．

step2：仮説の尤度計算 各地名単語列候補の各単語について，

・接続されたタグの数X

・タグ接続で飛ばされた文字数Y

を計算し，X－Yをその単語の尤度とする．XとYはそれぞれ，単語内で文字認識候補に一致した文字数，一致しなかった文字数を意味するので，以下，Xを「一致文字数」，Yを「不一致文字数」とも呼ぶ．次に，各単語について前後に接続すべき単語や街区の候補が存在した場合，各々，一致文字数Xに1を加える．これは，単語レベルで前後関係から見た位置的妥当性を，尤度に加えていることを意味する．地名単語列候補の尤度は，それに含まれる単語の尤度値（X－Y）の総和とする☆☆☆．

step3：検証による仮説の尤度補正 各地名単語列候補に含まれる文字タグ法での飛ばし箇所について，検証処理によりXやYの値の補正あるいは仮説の棄却を行う．

step4：判定 残った単語列候補の各単語について，Yが2以上のものを確信度が低いとして棄却する．そのうえで，地名単語候補列の尤度の値が最も大きな単語列を，読み取り結果として出力する．最大値を持つ候補が複数ある場合，判定不能で全候補を棄却とする．

なお，尤度の計算式，棄却や結果出力の条件などは，約100枚の画像（評価に使用した画像とは重複しない）での予備実験から定めた．

4.3 検証処理

3.2節に示した3種類の検証方式を，以下のような条件設定によって実装した．

下位候補検証： 文字認識候補の10位までに知識から推測される文字が存在すれば，その単語の一致文字数Xを＋1，不一致文字数Yを－1し，尤度を補正する．すなわち，検証した文字に対する文字タグが存在したのと同じ状態の尤度にする．

画数検証： 知識から期待される文字の画数が，それに該当する文字切り出し領域の文字認識候補の平均画数と大きく異なれば，単語を棄却する．「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」を棄却条件とした．

領域サイズ検証： 知識から期待される文字の数に，対応する文字切り出し領域のサイズが著しく不釣り合いな場合，単語を棄却する．具体的には，検証対象領域の前後の文字領域（ただし文字タグ法でタグが接続された文字領域）の平均幅を基準に，1文字あたり平均幅の2倍以上，あるいは1/3以下であった場合（たとえば，2文字が推定されれば平均幅の4倍以上あるいは2/3以下の場合）に，棄却する．なお，領域サイズ検証が複数単語にまたがって行われ棄却と判定された場合は，両方の

☆ ここでの「地名」とは，住所の都道府県名～町名までの部分とする．また，都道府県名，市区名，町名などの地名を構成する個々の要素を，「地名単語」と呼ぶ．

☆☆ ここでの街区とは，住所の丁目・番地・号の部分とする．

☆☆☆ 文字タグ法での尤度とは別に，一致文字数に基づく尤度を定義したのは，検証による尤度の補正を実験時に調整しやすくするためである．

表1 評価画像中の累積正解文字含有率
Table 1 Correct character containing rate of the experimental data.

認識上位候補数	1	2	3	4	5	6	7	8	9	10
累積正解含有率	0.717	0.770	0.781	0.798	0.808	0.813	0.817	0.819	0.820	0.826

表2 実験結果
Table 2 Experimental results.
(a) セットK（画像中の合計単語数 821）

認識候補数		1	2	3	4	5	6	7	8	9	10
検証なし	処理時間(ms)	80.4	85.2	89.2	91.0	93.5	97.4	100.9	102.2	105.6	109.1
	正読数	598	647	666	682	682	682	684	684	683	679
	誤読数	3	3	5	4	7	10	11	12	10	11
検証あり	処理時間(ms)	82.7	86.9	91.0	92.5	94.8	98.4	101.2	103.5	107.7	110.7
	正読数	644	671	681	688	683	682	683	681	679	676
	誤読数	4	3	3	3	6	12	13	15	12	11

(b) セットF（画像中の合計単語数 801）

認識候補数		1	2	3	4	5	6	7	8	9	10
検証なし	処理時間(ms)	108.4	121.8	131.6	142.1	151.5	161.0	169.0	177.3	186.4	195.7
	正読数	500	570	590	600	607	607	604	602	593	594
	誤読数	6	5	6	9	10	19	19	22	26	26
検証あり	処理時間(ms)	112.6	127.3	136.9	145.5	155.3	164.7	174.0	181.8	191.8	202.1
	正読数	560	604	609	609	608	606	601	599	589	588
	誤読数	9	5	7	10	12	20	20	22	25	25

単語とも棄却する．

なお，下位候補検証が成功した文字領域に対しては，画数検証と領域サイズ検証は起動しない．また，簡易検証に関連する条件は，尤度計算式などの決定（前節）と同じ予備実験を通して経験的に定めた．

4.4 実験方法

実験では，同一のパタン処理出力に対して，文字タグ法での仮説生成に使用する文字認識候補数を上位1位～10位まで変化させながら，知識処理の処理時間と読み取り精度を測定する．それを，検証を行った場合と検証を行わない場合☆で比較すれば，短い時間で高い精度を達成できるという仮説検証型アプローチの効果について考察できる．読み取り精度は，画像に記載された地名単語の正読数（正しく読んだ数）と誤読数（誤って読んだ数）によって評価する☆☆．

4.5 実験対象データの性質

実験対象の画像は，葉書の縦書き住所で，東京都国分寺市と静岡県富士市宛の画像，各400ずつを用意した．それぞれ，「セットK」，「セットF」と呼ぶ．住所の知識として単語辞書に登録する地名単語数は，それぞれ24と205である．セットFの方が単語数も多く，また類似の地名も多いので，知識処理の難度が高いと推測できる．実際の画像には，セットKで821，セットFで801の単語が記載されていた．パタン処理においては，文字領域の切り出しに石寺らの方式[16]を，文字認識に津雲の方式[17]を用いた．知識処理への入力となるパタン処理出力は，文字切り出しの平均多重度1.9☆☆☆☆であった．認識候補数と累積正解文字含有率の関係は，**表1**に示した．

パタン処理結果は，画像の状況や文字認識に使用する辞書の状況によって性能の変動が激しい．手書き文字列に対しては，その傾向が特に顕著であり，実験で使用したパタン処理技術の一般的性能を示すのは困難である．したがって，前述の文字切り出しの平均多重度と累積正解文字含有率は，実験に使用した800枚の評価画像に対しての実測値を提示した．ただし，手書きの住所に対する認識技術を検討した文献15)によれば，この正解含有率は現状の標準的な技術水準にあることから，実験データとして特に不適切ではないと考える．

4.6 定量的考察

実験結果として，まず**表2**に，仮説生成に使用した文字認識上位候補の数と知識処理フェーズの平均処理時間および読み取り単語数の関係をまとめる．

☆ 検証を行わない場合は，従来のボトムアップ型に相当する（1，2章参照）．

☆☆ 正読数，誤読数に，棄却した単語の数を加えると，全単語数となる．

☆☆☆☆ 正解文字に対する文字切り出し候補数の平均．

4.6.1 文字認識候補数と知識処理時間の関係

まず，仮説生成に使用する文字認識候補数と知識処理フェーズの処理時間の関係を考える．表2から，両セットとも，認識候補数が増えるに従って処理時間が増加していることが分かる．文字タグ法の効率の良さから，処理時間の爆発的増加は抑えられているが，仮説生成で処理する候補数の増加が探索空間を広げ，処理時間を増加させることが分かる．セットKとセットFで処理時間が大きく異なるのは，単語辞書に登録されている地名単語数の違いによると考えられる．辞書中の単語数が多ければ，文字タグ法で作成するタグ数が増加するため，仮説生成の時間が増加する．

4.6.2 検証処理の時間

検証の有無による処理時間の差は，両セットとも数msと非常に小さいものであった．知識処理時間全体に占める検証処理の割合は，最大でも，セットKで文字認識候補数が1のときの2.7%，セットFで文字認識候補数が2のときの4.4%であった．今回実装した3種類の検証方式は，方針どおり，非常に軽い処理であることが確認できた．

4.6.3 文字認識候補数と処理精度の関係

次に，認識候補数と処理精度の関係を考察する．実験セットにより難易度が異なるため読み取りの精度に差はあるものの，両評価セットに共通の傾向がいくつか見られる．まず正読単語数は，検証の有無にかかわらず，最初，認識候補数が増加するに従って増加するが，途中から減少してしまう．検証の効果は，認識候補数が少ない場合ほど顕著だが，認識候補数が多くなるとほとんどなくなるか，逆に悪影響となるケースもある．一方，誤読単語数については，検証の有無による差はあまりなく，認識候補数の増加に従って増加する．ただしこちらも，認識候補数が多いと，検証を行った方の誤読が多くなったケースがある．

正読単語数の変化については，次のように説明できる．まず，あまりに少ない認識候補数による仮説生成では，正しい仮説の生成に必要な情報が不足して，読み取り精度が低くなる．この問題は，ある程度の候補数を処理することで解消され性能のピークが訪れるものの，それ以上の認識候補数を処理すると，逆に誤った認識候補と知識が偶発的に一致して作られた仮説との競合が発生するので，読み取り精度が下がってしまう．誤読に関しては，認識候補数が増えれば，候補があげられる可能性は高くなるが，それが誤りである可能性もまた高くなるという説明ができる．

この結果から，少なくとも仮説を生成するパタン処理候補の数が少ないケースでは，検証処理によって読み取り性能が大きく向上することが分かった．すなわち，パタン処理の上位候補からの仮説生成と検証を適切に組み合わせると，高い処理精度を達成できることが示された．また，パタン処理の出力する候補を多く使って読み取り候補（仮説）を作成し正読率を上げるというボトムアップ型の試みは，あるピークを超えると性能劣化につながるおそれのあることが示された．この点については，後の定性的な考察でも議論する．

4.6.4 知識処理時間と読み取り精度の関係

表2だけでは知識処理時間と読み取り精度の関係が分かりにくいので，表から横軸に知識処理フェーズの平均処理時間，縦軸に正読単語数/誤読単語数をとり，**図7**に整理した．正読数を見ると，両セットとも，検証処理を行った場合の性能のピークが早く（短い時間で）訪れ，ピーク値も高いことが分かる．セットKでは，検証なしのとき100.9 msで684単語の正解に対し，検証ありは92.5 msで688単語であった．またセットFでは，検証なしの151.5 msで607単語に対し，検証ありでは136.9 msで609単語となった．これは，文字認識上位候補だけを使った仮説生成と簡易な検証の組合せが，多くの候補を使った全探索よりも処理時間と精度の面で優れた性能となったことを示している．

4.7 検証処理の効果

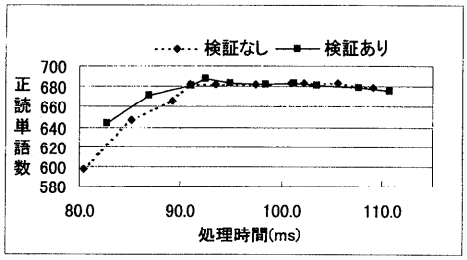
次に，各検証処理が最終出力にどれくらい影響したかを，**表3**に示す．同じ認識候補数で仮説生成した場合に，検証の有無で生じた差の原因となった検証をカウントしたものである．「正読＋」は正読を増やす効果があったもの，「誤読－」は誤読を減らす効果があったもの，「正読－」は正読を減らす悪影響があったもの，「誤読＋」は誤読を増やす悪影響があったもの，である．前の2つが性能改善，後の2つが改悪となる．結果は，実験セットごと，検証ごとにまとめた．以下，改善や改悪の実例を交えながら考察する．

4.7.1 下位候補検証の効果

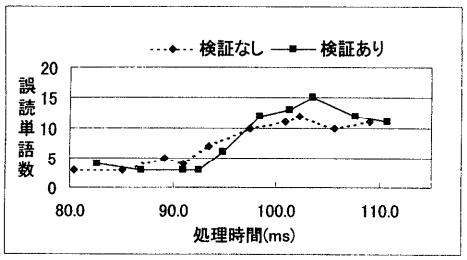
まず，認識候補数が少ないとき，すなわち短い処理時間ときの性能向上に大きく寄与しているのは，下位候補検証であることがはっきりと分かる．文字認識の上位候補のみによって作られた仮説を，下位候補によってうまく補完することができた．

下位候補検証による性能改善の典型的な例を**図8**に示す．図の左側が住所の画像，右上が個別文字切り出しと認識の候補☆，右下は読み取り文字列候補である．

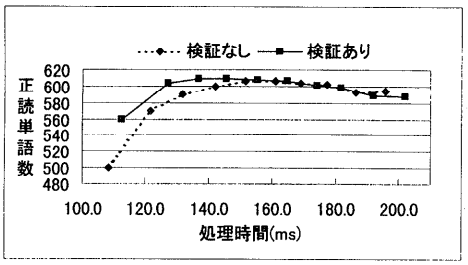
☆ 文字切り出し候補は(x,y)の形式で表現されているが，xは最小切り出し領域（セグメント）での位置，yは最小切り出し領域の統合数である．



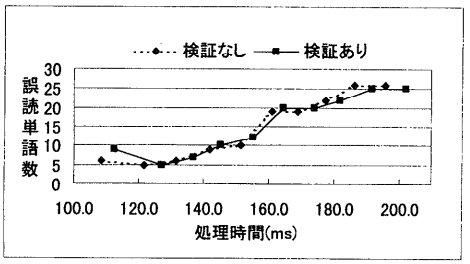
(a-1) 処理時間と正読単語数（セットK）



(a-2) 処理時間と誤読単語数（セットK）



(b-1) 処理時間と正読単語数（セットF）



(b-2) 処理時間と誤読単語数（セットF）

図7 処理時間と正読/誤読単語数の関係
Fig. 7 Relations between the processing time and the word reading accuracy in the experiments.

読み取り候補文字列に付いている「○」は，仮説生成時にタグが接続されたか，あるいは下位候補検証が成功したこと，「△」はタグが接続されなかったこと，を意味する．また，改善と改悪に直接関係した文字は「★」を付けて強調してある．

この例では，文字認識の上位5位までで仮説生成を行っている．検証を行わない場合，「日吉」という町名と「内藤」という町名が，それぞれ文字領域(7,1), (8,1)で2文字中の1文字ずつ認識候補にあり，競合となる．検証を行うと，下位候補検証により文字領域(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．下位候補検証による改善は，このような競合回避，あるいは検証により単語の不一致文字数Yが補正され，単語が棄却されなくなったケースがほとんどであった．

一方，下位候補検証で改悪となるのは，認識誤りの候補から作られた誤った仮説を下位の認識候補が補強し，その結果，誤読する，あるいは正解候補と競合状態になるケースであった．しかし，その頻度は小さかった．

表3 読み取りに変化を与えた検証処理の内訳
Table 3 Counts of verifications that cause changes in reading results.

(a) セットK

認識候補数		1	2	3	4	5	6	7	8	9	10
下位候補検証	正読+	41	22	10	6	3	3	2	1	1	0
	誤読−	0	1	0	0	0	0	0	0	0	0
	正読−	0	1	0	0	0	0	0	0	0	0
	誤読+	1	1	0	0	0	0	0	0	0	0
画数検証	正読+	6	4	5	2	1	0	0	0	0	1
	誤読−	0	1	1	0	0	0	0	0	0	0
	正読−	0	0	0	2	2	2	2	3	4	2
	誤読+	0	1	0	0	0	3	3	4	2	1
サイズ検証	正読+	0	0	0	0	0	0	0	0	1	0
	誤読−	0	0	1	1	1	1	1	1	0	1
	正読−	1	1	0	0	1	1	1	1	2	2
	誤読+	0	0	0	0	0	0	0	0	0	0

(b) セットF

認識候補数		1	2	3	4	5	6	7	8	9	10
下位候補検証	正読+	67	35	20	13	7	4	3	1	0	0
	誤読−	0	0	0	0	0	0	0	0	0	0
	正読−	4	2	3	2	2	0	0	0	0	0
	誤読+	6	3	3	2	2	2	1	1	0	0
画数検証	正読+	1	2	2	1	0	0	0	0	0	0
	誤読−	1	1	1	1	0	0	0	0	0	0
	正読−	2	0	0	3	3	4	4	1	1	3
	誤読+	0	0	0	0	0	0	1	0	0	0
サイズ検証	正読+	0	0	0	0	0	0	0	0	0	0
	誤読−	2	2	1	0	0	1	1	1	1	1
	正読−	2	1	0	0	1	1	2	3	3	3
	誤読+	0	0	0	0	0	0	0	0	0	0

4.7.2 画数検証の効果

画数検証は，セットKで，認識候補数が少ない場合に正読向上に一定の効果が見られた．それ以外では，改善と改悪が同等かあるいは検証が悪影響を与えている．改善では，1文字が複数の文字に分割され，その一方がたまたま地名の文字だったケースの棄却，あるいは，「国久保」「大久保」（セットFの町名）といった類似単語を，画数の差によりうまく判定できたケースが見られた．一方で，**図9**に見られるような，改悪ケースも見られた（図の見方は，図8と同様である

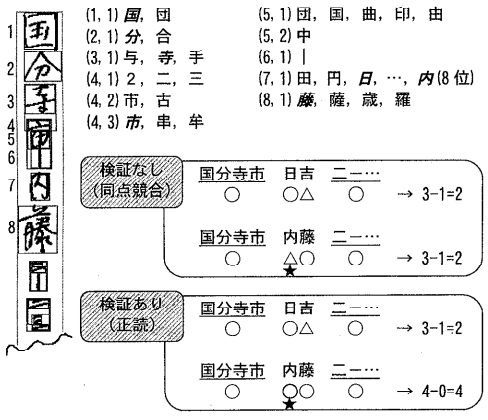


図8 検証処理による改善例

Fig. 8 An example of improvement by a verification process.

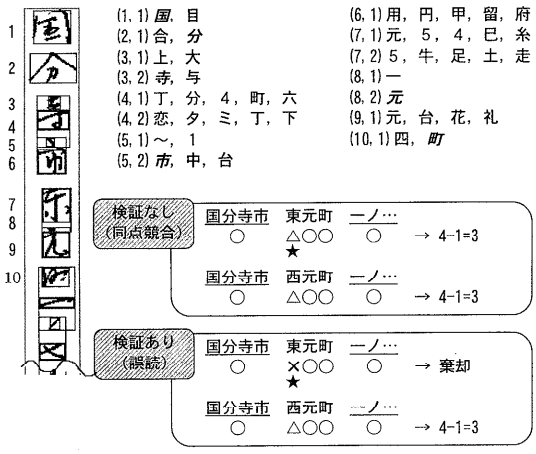


図9 検証処理による改悪例

Fig. 9 An example of a change for the worse by a verification processs.

が，加えて,「×」は画数検証や領域サイズ検証によって単語が棄却される原因となった文字を意味する).

この例では，画数検証によって，競合による棄却が誤読に変化した．文字領域 (7,1)「東」が略字で書かれているために正しく文字認識できない.「東元町」と「西元町」が候補にあがっても，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) の領域に対する認識候補の画数が小さく,「東」は検証による棄却条件に合致するので，結果として誤った「西元町」が選ばれてしまった．

この例に限っては，表記が「東」の略字であるため，文字認識候補の画数が少なくなったことが原因である．しかしこのほかでは，単語中の文字が 2 つに分割されてしまい，その片方だけを使って画数検証したために，正しい仮説を棄却する改悪例なども見られた．

画数検証は，文字の複雑さの代わりに画数を近似的に用いたものであるが，今回の実験から，近似ゆえの問題/限界を持つことが分かる．上記の略字の問題もその 1 つであるし，図 9 において同じ切り出し領域に「丁」と「町」,「恋」と「丁」のような画数の大きく異なる文字候補が出現していることも近似の限界を示すものである．

4.7.3 領域サイズ検証の効果

領域サイズ検証は，性能的に大きな改善をもらたすには至らなかった．効果が大きく現れなかった原因の 1 つとして，文字タグ法の尤度計算が，タグを飛ばして接続した際の文字数と文字切り出し領域の個数の整合を考慮していることがあげられる．つまり，極端に異常な大きさのブロックは，文字タグ法ですでに棄却されてしまっている．文字タグ法である程度離れた文字を誤って接続してしまった候補を棄却する改善例は見られたが，住所が乱雑に書かれており領域サイズが安定しない場合に，検証がマイナス効果になることもあった．

4.7.4 簡易検証の限界

簡易検証，特に下位候補検証は，仮説生成に使用する認識候補数が少ないときに有効に機能し，短時間で高い読み取り精度を得るのに大きく貢献することが分かった．

一方で，必ずしも検証が有効に機能しないケースもあった．今回実装した 3 つの簡易検証の特徴として，認識候補数が多くなるに従って検証による改善が少なくなり，それにともなって，改悪が増えるという傾向が見られた．特に，画数検証，領域サイズ検証でそれが顕著である．誤った読み取り候補が競合によって棄却されていたケースにおいて，簡易検証処理が他の対立候補（正解の場合もあるし，誤りの場合もある）を棄却するように働き，結果として残った誤読候補を出力してしまうという改悪が目立った．パタン処理の大量の候補から作られる誤った仮説を，今回実装した簡易検証だけでは排除しきれなかったものと解釈できる．

4.8 仮説検証型アプローチの有効性

4.6.4 項において，文字認識上位候補だけを使った仮説生成と簡易な検証の組合せによる仮説検証型アプローチが，多くの候補を使った全探索というボトムアップ型アプローチよりも，短い知識処理時間で高い読み取り精度を達成できたことを述べた．

これは，2.2 節で述べた本論文の主張（仮説検証型アプローチは短い処理時間で高い読み取り精度を実現する可能性を持つ）に沿った結果であるが，主張の実証としては必ずしも十分ではない．4 章の実験では，

知識処理フェーズに閉じた簡易な検証手法しか実現していないため，処理時間と読み取り精度の適切なバランスも知識処理フェーズに閉じた効果にとどまっている．通常，文字列読み取りの全体処理時間に占める知識処理時間の割合は，1割以下にすぎない．4.6.4 項での知識処理時間の比較によると，知識処理時間の短縮効果は1割程度であるから，文字列読み取りの全体処理時間に対しては1%に満たない効果ということになる．仮説検証型アプローチの優位性を実証するためには，今後さらに，パタン処理を再起動するタイプの検証方式も開発し，それを組み合わせた形での評価が不可欠である．

4 章の実験は，知識処理フェーズに限定した形で主張（仮説検証型の利点）を裏付けたものであるわけだが，パタン処理フェーズまで含めたときにも同様の主張が成り立つことを期待させる一面も示している．4.6.3 項で考察したように，知識処理がパタン処理から多数の候補（文字切り出し/認識の候補）を受け取ると，ボトムアップ型アプローチでは，知識処理の探索空間の拡大による処理時間の増大のみならず，偶発的な誤読の危険性も増す可能性がある．パタン処理フェーズまで含めてボトムアップ型と仮説検証型の比較を実施したときに，同様の現象は十分に起こりうるものであり，これが仮説検証型アプローチの性能を優位な方向へ導く．

5. おわりに

本論文では，仮説検証型の文字列読み取り知識処理方式が，短い処理時間で高い処理精度を達成しうるものだと主張した．また，その要素技術として，文字タグ法による仮説生成と，下位候補検証，画数検証，領域サイズ検証という3種類の簡易検証処理を提案し，それらを組み合わせて実装したシステムを試作した．これを手書き住所読み取りに適用した実験によって，知識処理フェーズに閉じた評価ではあるが，仮説検証型アプローチの有効性/可能性を示した．仮説生成と検証の組み合わせ方を，処理時間と処理精度のバランスという新しい観点から議論し，その有効性を示したことは，文字列読み取り知識処理の新たな視点の提示として意義を持つと考える．

4.8 節で述べたように，本論文で示した検証方式と実験では，仮説検証型アプローチの効果が知識処理フェーズの範囲内だけにとどまっている．これをパタン処理フェーズも含めた形に拡張することによって，仮説検証型アプローチの優位性をより明確に示すことが，今後の重要な課題である．

図3によれば，検証方式として，異なる文字切り出し方式を用いた文字切り出し処理（S2），文字切り出し結果の下位候補を対象とした文字認識処理（R2），異なる文字認識方式を用いた文字認識処理（R3），文字認識結果の下位候補を対象とした知識処理（K2），異なる情報を参照した知識処理（K3）などの可能性がある．本論文では K2・K3 のタイプに属する検証方式の一例を示したにすぎない．K2 や K3 タイプの検証方式でも，画数のような近似情報ではなく，密度やストローク数に着目するなど，検証方式はほかにも考えられる．さらに，S2・R2・R3 のようなパタン処理を再起動するタイプの検証方式も考えられる．この場合，知識から推測される文字（文字列）が分かった状態での実行となるため，従来とは異なる新しい文字切り出し/認識の方式が生まれる可能性もある．より精密な検証方式の考案・実装を目指していきたい．

また，処理時間に関する制約を無視したときに，同レベルの候補数/情報量を用いたボトムアップ型と仮説検証型とで，いずれがより高い読み取り精度を達成しうるのかは，現時点では，必ずしも明らかになっていない．図3を参照して述べるならば，ボトムアップ型の $(S1 + S2) \times (R1 + R2 + R3) \times (K1 + K2 + K3)$ と仮説検証型の $(S1 + a \cdot S2) \times (R1 + b \cdot R2 + c \cdot R3) \times (K1 + d \cdot K2 + e \cdot K3)$ の読み取り精度の上限比較の問題である．ボトムアップ型ではすべての箇所で詳細な解析を行うのに対して，仮説検証型では部分的にのみ詳細な解析を行うということから単純に考えると，一見，仮説検証型ではボトムアップ型の精度を超えられないとも思える．しかし，4 章の実験では，多数の不要候補が混じった広い空間を探索することは，誤った候補の組合せによる偶発的な誤読の危険性が増す☆という，ボトムアップ型に不利な現象を示した．このような視点からも，今後，より多様で精密な検証方式の考案・実現が待たれる．

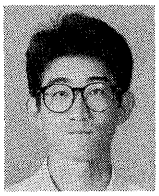
謝辞 本研究を進めるにあたって，パタン処理（文字切り出し，文字認識）の技術を検討してくださった，津雲部長をはじめとする日本電気株式会社 C&C メディア研究所パタン情報 TG の各位に感謝する．また，実験環境を準備していただくとともに，有益なコメントをいただいた日本電気株式会社産業オートメーション事業部，NEC ポスタルテクノレクス（株）の各位に感謝する．

☆ 多数の候補からの的確な選別や，多数の情報に対する的確な重み付けは，必ずしも容易なことではない．

参考文献

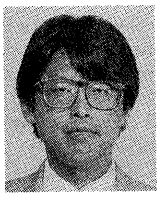
1) 高尾哲康，西野文人：日本語文書リーダ後処理の実現と評価，情報処理学会論文誌，Vol.30, No.11, pp.1394–1401 (1989).
2) 伊藤伸泰，丸山 宏：OCR 入力された日本語文の誤り検出と自動訂正，情報処理学会論文誌，Vol.33, No.5, pp.664–670 (1992).
3) 杉村利明：候補文字補完と言語処理による漢字認識の誤り訂正処理法，電子情報通信学会論文誌，Vol.J72-D-II, No.7, pp.993–1000 (1989).
4) 新谷幹夫，目黒眞一，梅田三千雄：認識情報及び単語・文節情報を利用した文字認識後処理，電子通信学会論文誌，Vol.J67-D, No.11, pp.1348–1355 (1984).
5) 長田一興，牧野 保，日高 達：日本語の文脈情報を用いた文字認識，電子通信学会論文誌，Vol.J67-D, No.4, pp.520–527 (1984).
6) 池田克夫，大田友一，上野恵美子：手書き原稿認識における語彙および構文の検定，情報処理学会論文誌，Vol.26, No.5, pp.862–869 (1985).
7) 木谷 強：手書き文書の文字認識結果に対する後処理方式，情報処理学会研究報告，91-NL-86-1 (1991).
8) 村瀬 洋，新谷幹夫，若原 徹，小高和己：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，電子情報通信学会論文誌，Vol.J69-D, No.9, pp.1292–1301 (1986).
9) 丹羽寿男，山本浩司，小島良宏，〆木泰治，丸野 進，萱嶋一弘：パターンと記号の統合化処理による文字認識，電子情報通信学会論文誌，Vol.J78-D-II, No.2, pp.263–271 (1995).
10) 佐瀬慎治，辻善 丈，津雲 淳：制限付き文字列読み取りの一検討，信学技報，PRU88-115, pp.49–56 (1988).
11) 仲林 清，松尾比呂志，津田伸生：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，第34回情報処理学会全国大会論文集，4E-8 (1987).
12) 下村秀樹，福島俊一，森 義和：手書き住所読取りにおけるパタン処理と連携した住所知識処理方式，第50回情報処理学会全国大会論文集，4D-1 (1994).
13) 福島俊一，下村秀樹，森 義和：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，情報処理学会論文誌，Vol.37, No.4, pp.500–510 (1996).
14) 下村秀樹：手書き住所読取りにおける街区住所知識処理方式，第51回情報処理学会全国大会論文集，4R-8 (1995).
15) 堤田敏夫，川又文男：住所読取りのための漢字認識複合化方式の一検討，1995年電子情報通信学会情報・システムソサエティ大会，D-201 (1995).
16) 石寺永記，西脇大輔，山田敬嗣：手書き住所読み取りのための文字切り出し方法，1995年電子情報通信学会総合大会，D-576, p.302 (1995).
17) 津雲 淳：手書き漢字認識，NEC技報，Vol.47, No.8, pp.76–81 (1994).

(平成10年3月25日受付)
(平成11年4月 1 日採録)



下村 秀樹（正会員）

1965年生．1993年東京農工大学工学研究科博士後期課程（電子情報工学）修了．同年，日本電気（株）入社．1998年7月よりソニー（株）．1990年本学会プログラミングシンポジウムにて山内奨励賞受賞．情報検索，文字認識知識処理，文書作成環境に興味を持つ．工学博士．



福島 俊一（正会員）

1958年生．1982年東京大学理学部物理学科卒業．同年，日本電気（株）入社．現在，同社ヒューマンメディア研究所主任研究員．高速で頑健な探索アルゴリズムの切り口から自然言語処理・情報検索分野の研究開発に取り組んでいる．情報処理学会，言語処理学会，人工知能学会，情報知識学会，情報科学技術協会，ACM 各会員．工学博士．情報処理学会平成4年度論文賞，第45回全国大会奨励賞，第53回全国大会優秀賞，平成9年度坂井記念特別賞受賞．

山内 俊史

1960年生．1983年神戸大学工学部電気工学科卒業．1985年同大学院工学研究科電気工学専攻修了．同年，日本電気（株）入社．現在，同社産業オートメーション事業部第二認識制御技術部エキスパート．郵便物自動読み取り区分機，文字認識方式の開発に従事．電子情報通信学会，視聴覚情報研究会各会員．