

# 文字誤認識の傾向を考慮したスペルチェッカーによる単語認識の精度向上

浅田 伸彦<sup>†</sup> 岩村 雅一<sup>†</sup> 黄瀬 浩一<sup>†</sup>

<sup>†</sup> 大阪府立大学大学院工学研究科 〒 599-8531 大阪府堺市中区学園町 1-1

E-mail: <sup>†</sup>asada@m.cs.osakafu-u.ac.jp, <sup>††</sup>{masa,kise}@cs.osakafu-u.ac.jp

あらまし 本稿では、環境中のあらゆる文字・単語を認識する「全方位認識」の実現のために、厳しい射影歪みに対応する手法について検討する。我々は実時間認識可能、レイアウトフリー、射影歪みに頑健といった3つの要件を満たすカメラベースの単語認識手法を提案している。この手法は斜め45度から撮影した文字を認識できる頑健さを持つが、前述の全方位認識に際しては、射影歪みに対する更なる頑健性が求められる。そこで本稿では、前述の単語認識手法の文字認識誤りをオープンソースのスペルチェッカーであるGNU Aspellを利用して補正する。その際、Aspellが持つ音素の類似性に基づくMetaphoneという仕組みを文字誤認識傾向に基くMetashapeに置き換える。提案手法を用いて実験を行った結果、文字が書かれた紙面に対して撮影角度が20度のときに単語認識精度には最大で約24%から約74%の向上が見られた。

キーワード 単語認識, 文字認識, 実時間処理, スペルチェッカー, カメラベースインターフェース

## Improvement of Word Recognition Accuracy with Spellchecker Based on Tendency of Recognition Error of Characters

Nobuhiko ASADA<sup>†</sup>, Masakazu IWAMURA<sup>†</sup>, and Koichi KISE<sup>†</sup>

<sup>†</sup> Graduate School of Engineering, Osaka Prefecture University

1-1 Gakuencho, Naka, Sakai, Osaka, 599-8531 Japan

E-mail: <sup>†</sup>asada@m.cs.osakafu-u.ac.jp, <sup>††</sup>{masa,kise}@cs.osakafu-u.ac.jp

**Abstract** In this paper, we discuss a method to cope with severe perspective distortion to realize “all-round recognition” of characters and words in the environment. We have proposed a word recognition method satisfying three requirements: ready for real-time processing, free from layout constraints and robust to perspective distortion. The method has the capability to recognize characters captured from the angle of 45 deg. However, in order to realize the all-round recognition, further robustness is required. Thus in this paper, we introduce the GNU Aspell to rectify the recognition results based on the tendency of recognition errors. For the sake of that, we introduce Metashape based on the recognition errors instead of Metaphone of the Aspell based on phonemes. In the experiments, we found that the proposed method improves the recognition rates for words captured from the angle of 20 deg. from about 24% to about 74% at largest.

**Key words** Characters Recognition, Word Recognition, Real-Time Processing, Spellchecker, Camera-Based Interface

### 1. ま え が き

近年、カメラを用いた文字認識システムが注目を集めている。有望な応用としては、カメラと文字認識装置、翻訳装置を組み合わせた「翻訳カメラ」がある。

また更なる応用として、全方位カメラを使用した文字の全方位認識が挙げられる。全方位カメラを利用して、街中の文字を全て認識することができれば、視覚障害者に対する支援や、全

方位認識中に健常者に対しても外国でコーヒーショップがあれば知らせるなど街中の文字情報から利用者にとって有益な情報のみを抜粋して提供することなどが可能なるため、有望な応用であると考えられる。

辻らは文献[1]においてこのような応用を実現するための手法を提案している。この手法は実時間認識可能、レイアウトフリー、射影歪みに頑健といった3つの要件を満たすカメラベースの単語認識手法である。特に射影歪みに対する頑健性では、

斜め 45 度から文字を撮影しても正面から撮影した場合と同程度の認識性能を持つ。そのため、前述の全方位認識に適した認識手法といえる。しかし、全方位認識に際しては、文字が書かれた紙面に対する撮影角度をコントロールできないため、例えば、紙面との角度が 20 度であっても文字が認識できるといった射影歪みに対する更なる頑健性が求められる。

そこで本稿では、全方位認識に際して生じる厳しい射影歪みに対応するため、文献 [1] の手法の文字認識誤りを単語辞書を利用して訂正する手法を検討する。文献 [1] の手法は単語認識の際に辞書を使用していないため、これによって認識性能の向上が期待できる。単語辞書の利用方法として、本稿では文字認識の誤り傾向を反映したスペルチェッカーを導入する。

スペルチェッカーとしては、オープンソースの GNU Aspell を利用する。Aspell は入力文字列が与えられた際にスペルチェックを行い、スペルに誤りがあれば修正候補を提示し、ユーザーに単語修正の機会を与える。具体的な処理は、辞書に登録されている単語（英語の場合、約 13 万語）から与えられた文字列と編集距離の小さい上位 10 単語を正解単語候補として出力する。その際、約 13 万もの単語との編集距離を求めている計算時間が膨大になってしまい、実用に耐えない。そこで Aspell では計算時間を短縮する工夫として Metaphone というアルゴリズムを用いている。Metaphone は辞書内の同じ音素で構成される単語をグループ化するための手法である、辞書探索の際には、入力文字列と同じ音素で構成される特定のグループのみを探索対象とすることで、誤り修正能力を保ったまま計算時間の短縮を実現している。

このようにスペルチェッカーを用いることで単語認識の精度は向上すると考えられるが、Aspell で用いられている Metaphone は音素を基にした単語のグループ化手法であり、音素情報は文字認識とは直接関連がない。そこで本稿では、音素情報ではなく、文字認識誤りの傾向に基づいた文字の形状情報を利用して単語をグループ化することを提案する。ここではこの形状情報によるグループ化を Metashape と呼ぶこととする。具体的には、Aspell で使われている文字の音素による辞書を、前実験で調査した文字の形状類似による辞書に差し替えることで改良を施す。提案手法の有効性を確かめるため、Metashape を適用した改良型 Aspell を辻らの手法と組み合わせ、認識結果を調査する実験を行った。

以降は、2 節で辻らの手法と Aspell について解説し、3 節で提案手法について解説し、4 節で実験、5 節でまとめとする。

## 2. 既存手法

本節では本手法で利用した既存の単語認識手法である辻らの手法 [1] と既存のスペルチェッカーである GNU Aspell について解説する。

### 2.1 辻らの手法

辻らの手法 [1] は、白い背景に黒い文字で書かれた英数字を認識対象として想定している。まず初めに文字認識処理によって対象内の黒画素連結成分を文字候補として抽出する。アフィ

ン不変量を利用したアフィン不変なテンプレートマッチングを行っている。テンプレートマッチングを実現するためには様々なテンプレートが必要なため、認識対象である大文字と小文字のアルファベットと数字の合計 62 字種を 100 フォント分登録している（本稿の実験では 1 フォントのみ）。また、生成型学習により、1 字種当たり 9 種類（劣化なし 1 種類を含む）の劣化画像を生成したため、10 フォントでは合計 5,580 枚、100 フォントでは合計 55,800 枚の参照画像をデータベースに登録している。

単語認識処理では、最初に画像中のどの領域が 1 つの単語であるのかを推定する。図 1 のように文書画像にある程度のぼかしを掛けて二値化すると、隣接する文字同士が結合し、スペースで区切られた部分のみが分割したままの状態になる。よって、ぼかし画像の連結成分を抽出することで単語の領域を推定することが出来る。そして元の文書画像とぼかし画像の座標を照らし合わせて各文字がどの単語領域内に含まれるかを調べ、それぞれの単語を構成する文字を決定する。図 1 を例に挙げると、単語領域 2 から抽出される文字は (1)(2)(3)(4)(5) である。この段階での「文字」は先の文字認識処理によってクラスレベルでの識別が終了した状態であり、未だ各文字ごとに複数の字種の候補を持つ。今回の説明では、文字 (1) のクラスには ' M ' と ' W ' の 2 つ、文字 (3) のクラスには ' d ' と ' p ' の 2 つの字種を含み、それ以外は 1 字種のみで構成されたクラスとする。次に、単語ごとに文字の並び順と字種を推定する。情景中の文字列には様々なレイアウトが存在するが、いずれも人に読んでもらい、誘導するためにデザインされたものであるため、文字の並びや向きが急激に変化することはないと仮定できる。辻らの手法ではこの仮定を用いて単語内の文字の並びと字種を推定する。

単語内の文字の並び順の推定について述べると、例えば図 1 の単語領域 2 の単語を“ (4)(5)(1)(2)(3) ”などと途中で急に離れた場所の文字を読むようなケースは仮定に当てはまらない。辻らの手法では尤もらしい並びを求める方法として、単語内に含まれる文字を最短で 1 度ずつ通るものを選択、つまり最短経路問題をダイクストラ法で求める。経路は文書画像上の各文字を直線で結んだものであり、コストは文字間のユークリッド距離を意味する。結果としては“ (1)(2)(3)(4)(5) ”およびその逆順“ (5)(4)(3)(2)(1) ”が最短経路として得られる。次に、求めた順番通りに各文字の字種の候補を組み合わせることで単語の候補を挙げる。

次に文字の上方向について考える。ここでは「単語は左から右に読む」というルールを用いる。単語領域 2 の単語を例に取り、考え方を図 2 と図 3 で示す。“ Media ” のように単語を文字番号 (1) から順に読むとすると、図 2 のように 1 文字目から 2 文字目への方向が右方向と仮定できる。すると上方向はそれと直角な方向と仮定できる。上方向と 1 文字目の候補の角度との差が小さいほど尤もらしい文字であると言えるため、文字の上方向の候補は単語を読む 2 通りの経路にそれぞれ垂直な 2 方向に限定される。具体的には ' M ' と ' W ' について考える際に、図 2 のように単語を (1) から (5) の順で読むと、単語の上

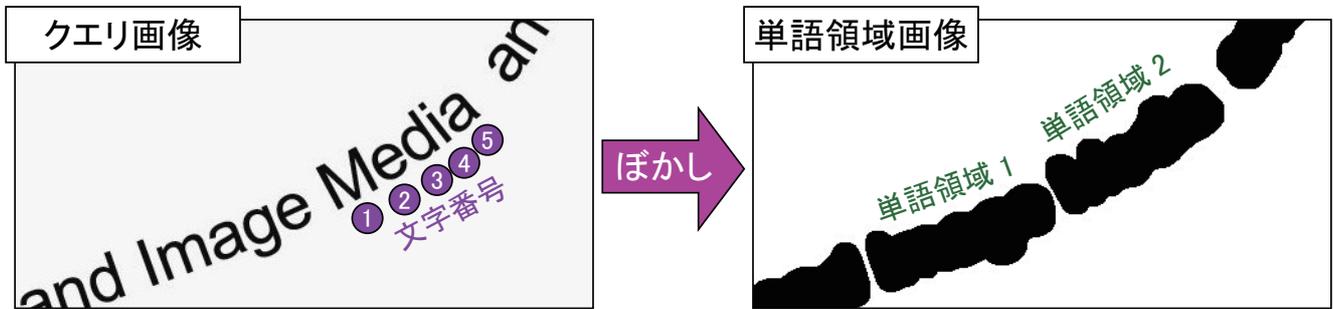


図 1 辻らの手法の処理概要 (文献 [1] より引用)

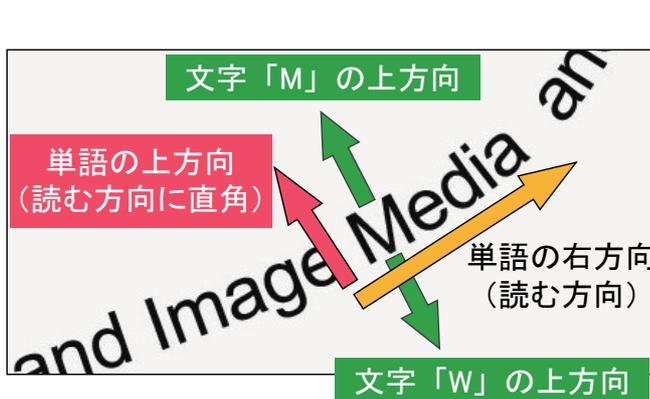


図 2 辻らの手法の最終的な字種決定 (文献 [1] より引用)

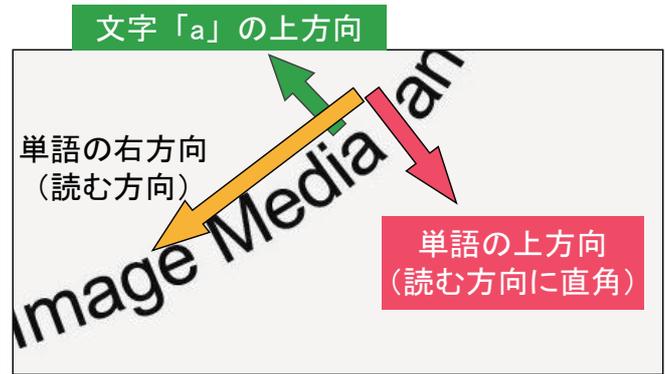


図 3 辻らの手法の最終的な字種決定 (文献 [1] より引用)

方向は 'M' の上方向と等しくなり、また 'M' の上方向は 'e' や 'a' の上方向とも等しくなる。これに対し、図 3 のように (5) から (1) の順に読む際には、単語の上方向は 'W' の上方向と等しくなるが、'W' の上方向は 'e' や 'a' の上方向と異なるため、(1) は 'M' である方が尤もらしいとして、(1) は 'W' ではなく 'M' と推定される。

表 1 辻らの手法で同一クラスに統合された字種。1つの欄が1クラスに当たる。

0 O o	6 9	7 L	C c	E m	I l	N Z z
S s	V v	W w	b q	d p	n u	

辻らの手法では、このような方法で、単語認識を行っているが、単語認識の段階では、基本的には文字認識での結果を組み合わせただけで、独自の新たな情報を用いての認識はなされていない。このため、文字認識処理で誤認識が発生した場合には、単語認識処理でそれに補正をかけることはできず、文字認識での誤認識が、そのまま引き続いて単語認識結果にも影響を与えていた。本稿では、これに対処するため、単語認識側で新たな処理を加え、結果を補正することを考える。

## 2.2 Aspell

Aspell は GNU ソフトウェアシステムの標準スペルチェッカである。Aspell はオープンソースとして提供されている。英語の場合、アルファベット 26 文字からなる文字列を対象とし、入力文字列が与えられた際に、辞書を参照し、まず、その文字列が Aspell の辞書内に存在するかどうかを検索する。その文字列が存在する場合には Aspell はスペルミスがないと判定する。文

字列が存在しない場合には、その文字列にはスペルミスがあるとして、入力文字列と似ている単語を修正候補として出力する。具体的には、Aspell の内部処理は大きく分けて、Metaphone と編集距離の 2 つのアルゴリズムからなる。Aspell では入力文字列が与えられた際に、その文字列と辞書内の単語の編集距離を求め、編集距離の小さい順に上位 10 単語を単語の修正候補とする。しかし、この際、Aspell には約 13 万もの単語が辞書登録されており、辞書内の全ての単語との編集距離を計算しては膨大な時間がかかってしまい、実用に耐えない。これを防ぐために Aspell では以下で述べる Metaphone を用いる。また、Aspell での編集距離では、スペルチェックの精度を高めるために、削除、挿入、置換といった処理以外に隣接した文字同士の交換も処理として加えている。また出力結果の上位 10 単語の内、どれが正解単語かは最終的にはユーザーに目視で確認させる仕組みになっている。

### 2.2.1 Metaphone

Metaphone では、入力文字列と辞書登録済みの単語を文字で構成されたものから音素記号で構成されたものに変換することで文字列同士をグループ化する手法である。具体的には事前にどの文字がどのような音素記号で表されるかを示す、辞書を作成しておき、入力文字列が与えられた際には、その辞書を参照することで、文字列を音素記号列に変換している。

この時、例えば y や i のような文字は同じ音素記号で表されるため、全ての音素記号のクラス数は全ての文字のクラス数より少なくなる。これにより、Smith と Smyth といった単語も音素の上では同じ記号列で表されることとなり、発音が類似し

ているものの綴りが異なる複数の文字列が類似の音素記号列で表されることとなるため、文字列のグループ化が可能になる。

Aspell ではこのことを利用して、編集距離計算の際には、入力文字列に Metaphone を適用し、得られた音素記号列と同じ音素記号列で表される単語のみを辞書から抜粋し、それらのみと距離計算を行うことで探索範囲を削減し、それに伴う計算時間の短縮を実現している。

### 3. 提案手法

Metaphone は文字の音素に注目し、多数ある文字のクラスをより数の少ない音素のクラスに変換し、クラス数を削減する手法であった。しかし、文字認識を考える上では、音素情報は関連性が薄く、あまり有効ではないと思われる。そこで本節では、文字のクラス数の削減を行う際に、音素情報を扱うことをやめ、代わりに文字形状の類似情報を扱うことを考える。Aspell は、クラス数の削減さえ有効に機能すれば、削減方法が異なっても有効に機能すると考え、より文字認識に適したクラスの削減方法として、文字認識の誤り情報を基に形状の似た文字同士は同じクラスとして扱い、全体のクラス数を削減することを考える。ここではこの手法を Metashape と呼ぶこととする。本稿では、この Metashape を組み込んだ改良型 Aspell を提案する。

#### 3.1 Metashape

例えば、辻らの手法では、アフィン変換に対応していることもあり、'N' と 'Z' や 'L' と 'V'、'p' と 'd' などの間にしばしば相互に誤認識が起こる。このような一定以上の確率で誤認識が発生する文字同士を一つのクラスで表すことにより、形状の類似によるクラス数の削減を行う。また形状の類似は、文字の対一対応に限らず、複数の隣接した文字が繋がった結果、別の文字と類似した形状になるといったことも起こり得るが、今回用いた手法では、簡単のため、形状の類似は文字の対一対応についてのみ考え、複数の文字が繋がって一つの文字になるケースについては考慮しない。尚、今回は字形が明らかに類似している場合のみならず、角度のきつい斜めから撮影した場合の射影歪みをアフィン歪みに近似しきれないところで起こる誤認識も対象とする。

#### 3.2 編集距離のコスト

提案手法では文字の対一対応の置換のみを考慮しているため、今回用いた Aspell 内の編集距離では置換のみを許容し、置換以外の削除、挿入、隣接した文字同士の交換は起こらないようにした。

### 4. 実験

提案手法の有効性を確認するため、辻らの手法と組み合わせ、カメラで撮影した文書内の単語認識精度を測る実験を行った。具体的には、単体で出力された辻らの手法の単語認識結果に対して提案手法を用い、スペルチェックにかけることで、補正した単語認識結果出力し、その認識精度を測定した。



図 4 前実験の撮影風景



図 5 本実験の撮影風景

#### 4.1 前実験

本実験を行う前に、どの文字同士を同一のクラスにするかを定めるため、異なる文字同士の誤認識傾向を調査する前実験を行った。前実験では、図 4 のような白い背景の紙面に黒文字で大小 3 通りのサイズの同一文字が書かれた紙面を対象とし、紙面から 20 度の角度から一定時間、動画像の撮影を行い、全フレーム中のどの程度のフレームにおいて文字の誤認識が発生するかを測り、誤認識率を測定した。提案手法では、この前実験の結果から confusion matrix を作成し、何%以上、文字間の誤認識が発生すれば、それらの文字を同一のクラスとするかによって、最終的な全クラス数を決定する。

#### 4.2 実験条件

図 6 のような白い背景の紙面に黒文字で書かれた、also, captured, enables, Even, real の 5 単語について、それぞれ 45 度、20 度の角度から、一定時間、動画像の撮影を行い、全フレーム中のどの程度のフレーム数において単語の誤認識が発生するかを測り、認識率を測定する。辻らの手法は 45 度の角度までは問題なく認識が可能であるが、全方位認識を行うためには 45 度では心もとない。そこで本実験では 20 度からの撮影での実験も行うことで、提案手法の有効性をより深く調査する。

表 2 クラス分類表 (20)

A	B,Q	C	D	E,M
F	G	H	I,L	J
K	N,U,Z	O	R	S
T	V	W	X	Z

表 3 クラス分類表 (11)

A,F,I,J,L,V	B,D,G,P,Q	C
E,M,W	H,N,U,Z	K,X
O	R	S
T	Y	

表 4 クラス分類表 (1)

A,B,C,D,E,F,G
H,I,J,K,L,M
N,O,P,Q,R,S,T
U,V,W,X,Y,Z

表 5 紙面に対する撮影角度と実験結果 [%]

	also		captured		enables		Even		real	
	20 °	45 °	20 °	45 °	20 °	45 °	20 °	45 °	20 °	45 °
辻ら	14.7	28.8	89.9	100	24.3	29.2	77.2	100	48.7	42.7
提案手法 (20)	35.8	90.8	98.9	100	68.0	80.6	77.2	100	49.4	42.7
提案手法 (11)	35.8	90.8	98.9	100	72.9	80.6	77.2	100	48.7	42.7
提案手法 (1)	35.8	90.8	98.9	100	74.1	84.5	77.2	100	49.4	42.7

用いた計算機は CPU が Intel Core i7 3.07GHz、メモリが 3GB で、Point Grey Research 社の Flea3 120fps のカメラを用いて撮影を行った。本実験では、上記環境の下、図 6 に示す 5 つの条件において認識率の比較を行った。各条件はそれぞれ、辻らの手法単体、提案手法のクラス 20 の場合、クラス数 11 の場合、クラス数 1 の場合の 4 つである。提案手法の同一クラス内の文字の内訳をそれぞれ表 2～表 4 に示す。' B ' と ' Q ' など形状的に類似していないように思われる文字が同じクラスとして扱われているのは、辻らの手法と Aspell の文字の取り扱い方の違いによるものである。辻らの手法は大文字小文字を合わせたアルファベット 52 字種と数字 10 字種の合計 62 字種を対象として、それらのアフィン変換の関係からクラスを作成しているのに対し、Aspell では大文字のアルファベット 26 字種に対してのみクラス作成を行っている。この時に、辻らの手法では小文字の ' b ' と ' q ' がアフィン変換の関係にあるため、同じクラスとして処理されるので、Aspell でもそれを引き継ぐため、' B ' と ' Q ' を同じクラスとした。その他の一見形状が類似していない文字が同じクラスになっているケースについても同様である。

#### 4.3 実験結果

実験結果を表 5 に示す、また単語認識結果の例を図 6 に示す。提案手法を辻らの手法と比較した場合、どのクラス数においても、認識率の向上が見られた。また撮影角度が 20 度の場合と 45 度の場合の双方で認識率が向上した。単語によって結果は異なった。

提案手法を用いて、認識率の向上が見られたことから、スペルチェックを用いた単語認識の精度向上は有効であると考えられる。個別の認識結果に目を向けると、also の 45 度での辻らの手法と提案手法 (20) では、約 62% 精度向上した。これは辻らの手法では大文字の ' I ' と小文字の ' l ' といった形状の類似していた文字を上手く認識できていなかったため、also が also と誤認されてしまうことが頻発していたのが、スペルチェックの辞書利用により改善されたためである。また captured と real の認識率の差など、単語の種類によっては認識率に大きな差が見られた。これは captured の誤認識結果として頻出した captured には、スペルチェック後の単語候補に captured と同程度の編集

距離を持つものが他になく、結果として captured に一意に決定されていたのに対し、real の誤認識結果として頻出した real には、編集距離が同程度に小さい単語が real 以外に、read など複数あり、辻らの手法での認識結果を単純にスペルチェックにかけるだけでは、real に上手く補正されず、その多くは別の単語に補正されていた。また特殊な誤認識のケースとして、also の誤認識結果として asia が出力されていることがあった。これは辻らの手法が、単語の向きと字種の決定の際に、個別の文字の認識時に推定できる上方向の情報を利用してしたが、also を構成する文字の内、a 以外の文字は全て文字の上方向を一意に決定できない文字である。このため、a の上方向の情報だけでは、上手く単語の上方向を決定することができず、まず初めに also が also と誤認識され、次に上方向の誤決定により、osIa という文字列として処理され、この文字列にスペルチェックをかけたことにより、osIa が asIa と変換され asia になったのだと考えられる。これら real が read に誤認識される問題、also が asia に誤認識される問題には、スペルチェックの処理を辻らの手法での単語認識結果の決定前に組み込むことで対処可能だと考える。

現在の実装では、辻らの手法の単語認識結果出力後に一度だけ、スペルチェッカーを用いているため、辻らの手法での第一候補についてのみスペルチェックを行っており、これは辻らの手法での第二候補にスペルチェックをかけた方が有用な結果が出力されるかもしれないことを想定すると効率が悪い。このことからこれらの問題に対処するために、スペルチェッカーを辻らの手法の処理後に単独で利用するのではなく、辻らの手法内部に組み込み、第二候補以降にもスペルチェックを加え、その結果たちの中から最終的な単語認識結果を出力するのが有効であると考えられる。

#### 5. まとめ

辞書を用いない単語認識では、単語認識結果が文字認識結果に依存していた問題に対処するため、スペルチェッカーを用いた単語認識の精度向上について提案、実験を行った。提案手法では文字の形状類似により文字をグループ化することを考えた。実験の結果、認識精度に向上が見られ、本手法の有効性が示さ

	撮影画像	辻ら	提案手法(20)	提案手法(11)	提案手法(1)
20度	also	also,aiso,o8ia	also,AIDs,olia	also,AIDs,olia	also,AIDs,olia
	captured	captured,capturod	captured,capturod	captured,capturod	captured,capturod
	enables	enables,s6lbane	enables,olivine	enables,seining	enables,seining
	Even	Even,u0ve	Even,nave	Even,nave	Even,nave
	real	real,real,i86r	real,read,ir's	real,read,ir's	real,read,ir's
45度	also	also,aiso	also,AIDs	also,AIDs	also,AIDs
	captured	captured	captured	captured	captured
	enables	enables,selbane	enables,olivine	enables,seining	enables,seining
	Even	Even	Even	Even	Even
	real	real,real	real,read	real,read	real,read

図 6 単語認識の結果例

れた。しかし、スペルチェックの際の誤認識や、スペルチェック前の他の単語候補の破棄などにより、100%の精度が出る訳ではないので、さらなる改良が必要である。

## 6. 今後の課題

今後の課題としては、also の文字順が反転してスペルチェックにかけられ asia になる問題に対処することがある。このために、単語の上方向を推定する方法の改善が求められる。また real が read に誤認識される問題に対処するため、辻らの手法の単語認識結果出力後に、スペルチェックを行うのではなく、文献 [2] の手法のように、辻らの手法の内部にスペルチェッカーを組み込みが必要である。また今回はある一文字が別の一文字にご認識されることのみを扱ったが、実際には複数の文字が繋がって一つの文字と似た形状になることも考えられるので、一対一対応以上の拡張が必要である。また Aspell では英語の場合アルファベット 26 文字にしか対応していないため、英数字の文書を対象とする場合には、Aspell の数字対応も課題となる。

謝辞

本研究の一部は科研費補助金 (21700202) ならびに科学技術振興機構 CREST の補助による。

## 文 献

- [1] 辻 智彦, 岩村雅一, 黄瀬浩一, “リアルタイム単語認識技術を利用したカメラベース情報取得システム,” 電子情報通信学会技術研究報告, vol.109, no.418, pp.51-56, Feb. 2010.
- [2] J. Weinman, E. Learned-Miller, and A. Hanson, “Fast lexicon-based scene text recognition with sparse belief propagation,” Proc. ICDAR2007, pp.979-983, 2007.