

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

辻 智彦[†] 岩村 雅一[†] 黄瀬 浩一[†]

[†] 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: tsuji@m.cs.osakafu-u.ac.jp, {masa,kise}@cs.osakafu-u.ac.jp

あらまし ユーザーインターフェースの利便性に関わる要素としては素早い応答や直観的な操作性が挙げられる。我々が既に提案しているカメラベース文字認識手法は実時間処理が可能、射影歪みに頑健、文書レイアウトに依存しない、という特徴を持つ。我々はこの手法を利用することで新たな対話型インターフェースが実現するのではないかと考えた。本稿ではカメラで撮影した環境中の文字を認識して関連した情報を提供するシステムを提案し、ノートパソコン上で実時間で動作するデモシステムを紹介する。システムには 1 文字単位の認識結果から単語を推定する処理が必要なため、有効な単語認識手法についても述べる。提案手法ではサーバーを用いた場合 1 秒間に約 42 単語の認識が可能である。キーワード 文字認識, 単語推定, 実時間処理, リンクアンカー化, カメラベースインターフェース

A Camera-Based Information Acquisition System Using a Technique of Real-Time Word Recognition

Tomohiko TSUJI[†], Masakazu IWAMURA[†], and Koichi KISE[†]

[†] Graduate School of Engineering, Osaka Prefecture University

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

E-mail: tsuji@m.cs.osakafu-u.ac.jp, {masa,kise}@cs.osakafu-u.ac.jp

Abstract In order to realize a comfortable user interface, rapid response and intuitive operation are required. The camera-based character recognition method where we have already proposed has some advantages, which are ready for real-time processing, robust to perspective distortion, and free from layout constraints of documents. We think that there is a possibility to realize a new interactive interface by applying it. In this paper, we propose a system which recognizes characters captured with a camera and offers related information. We demonstrate a system which works on a laptop PC in real-time. Since it needs a process that recognizes words from characters recognized separately, we propose an effective word recognition method for the system. Our proposed method achieved to recognize about 42 words per second on a server.

Key words Characters Recognition, Words Estimation, Real-Time Processing, Making to Link Anchor, Camera-Based Interface

1. はじめに

ユーザーにとって便利なアプリケーションの条件とはどのようなものか。情報検索システムを例に挙げると、ユーザーの要望に応じた情報を素早く提示することは不可欠である。我々はこれまでにカメラベース文字認識を利用した新たな対話型アプリケーションの可能性を示している [1]。提案システムの概観は図 1 に示す通りであり、このシステムは撮影した環境中の文字を認識し、文字列にリンクしている情報を提供する。任意の文字列を指定してリンク先の情報にアクセスすることもでき、これを我々は「環境中テキストのリンクアンカー化」と呼ぶ。

提供される情報はテキストによるものに限らず、画像や音声といったマルチメディアのサービスも可能であり、近年多機能化している携帯電話のように新たなサービス配信ツールとしての発展が期待できる。

このアプリケーションのベースとなる我々の認識手法 [2] [3] では実用性を考え、

- (1) 実時間処理が可能である
- (2) 射影歪みに頑健である
- (3) 文字の配置に依らない認識が可能である

といった 3 要件を満たす。

また、我々はこの手法を改良し、認識精度を向上させた [4]。



図 1 提案システムの概観。(A ~ F は本文に対応する。)

しかし、我々の認識手法では要件 (3) を満たすために 1 文字単位で認識処理を行うことから、このままでは単語が認識できない。単語単位でリンクアンカー化を行うためには文字単位の認識処理で得られたクラスの中から尤もらしい字種を選択し、それを組み合わせて単語を推定する必要がある。

本稿では実時間文字認識手法の改良版 [4] をベースにした情報取得アプリケーションおよび実用性を考慮した単語認識手法を提案する。アプリケーションの実装例としてデモシステムを紹介し、実験によって単語認識手法の有効性を示す。

2. 提案システム

2.1 提案システムの有用性

本節では実時間文字・単語認識技術を利用した情報取得アプリケーションを提案する。雑誌の記事や街頭の看板など、環境中には至る所に文字が存在し、それぞれが目的地への経路や商品の宣伝のような何らかの意味を持った情報を人に伝えている。そのような環境中文字をカメラで撮影し、実時間でコンピュータに認識させれば様々なサービスが見込める。例えば翻訳機能を組み合わせた「翻訳カメラ」[5] は非母国語で書かれた看板の内容を知りたいときに便利である。また、撮影した文字の中から重要な情報のみを音声で案内する装置があれば視覚障害者の新たな補助ツールとなり、非常に有用である。既に情景中から有用な情報を得る方法として QR コードを用いる方法が普及しているが、これにはあらかじめ QR コードを媒体に貼りつける手間やスペースが必要となる。それに対して、情景中の文字を用いる提案システムでは特別な準備が必要なく、気軽に使えるという利点がある。実世界の単語が各種サービスとリンクしていることから、このシステムを我々は「環境中文字列のリンクアンカー化」と呼ぶことにした。

2.2 デモシステムの概要

ここで、提案システムのデモ装置の概要を説明する。装置の概観を図 1 に示す。システムは市販のノートパソコンと小型カメラで動作し、持ち運んで使用できる。対象となる文書は白紙に黒色で文字が印刷されたものであり、レイアウトは任意である。カメラで用紙を撮影すると以下の処理や出力が毎フレーム、リアルタイムで行われる。

まず、キャプチャ画像 (図 1 の A、以下同じ) から文字領域

が抽出され、文字が認識される。抽出された文字領域は画面上では緑色で表示され、認識結果は 1 文字ごとに領域の中心に重畳される (B)。射影歪みに頑健な文字認識のため、斜めからの撮影にも対応し、撮影で生じた文字の歪みのパラメータを文字認識時に推定し、結果画像に歪みを反映することができる。例えば、正面から撮影された時よりも細長く映っている文字の上には同じ度合いだけ細くなった結果文字が重畳される。次に提案手法によって単語領域が抽出され、キャプチャ画像中に含まれる単語が求められる。画面上では単語領域が紫色で表示される。この処理で得られた単語が情報検索のキーワード、つまり「リンクアンカー」であり、あらかじめリンク付けしておいた各種情報にアクセスできる。

本デモシステムでは、ユーザーに提供するサービスの例として翻訳、画像表示、音声案内の 3 種類を実装した。翻訳の機能とは認識した英単語を英和辞典サーバーに送信し、訳語をコンソール上に表示 (C) するものである。画像表示については国名の単語に国旗と観光名所の画像をリンクさせ、カメラで撮影すると画像ウィンドウ (D) が開くようにした。同様に、音声案内の例では動物の単語にその動物の鳴き声をリンクさせた。

同一の単語でもフォントの違いによって異なるリンクを貼ることもでき、本デモでは例として Arial の場合のみ動物の単語を撮影すると鳴き声の代わりに写真を提示するようにした。フォント別サービスの応用例としては強調して書かれた文字を見つけたときにユーザーに知らせる機能などが考えられる。

キャプチャ画面に含まれる全てのリンクアンカーの情報を提示することもできるが、場合によっては雑多な情報をユーザーが受け取ることになってしまうため不便である。そこで、本システムではユーザーにとって有益な情報のみを取り出せるよう、任意の単語が選択できるモードを用意した。キャプチャ画面中央にカーソルが表示 (E) されるので、これを任意の単語領域に合わせて選択する。選択した単語領域は色が強調されたり別ウィンドウに文字列が表示 (F) されたりする。クリックしてサービスにアクセスできる他、単語がカーソルに重なると自動的にアクセスすることもできる。

3. 文字認識手法の概要

今回、提案システムで用いたカメラベース文字認識は我々が提案した手法 [4] である。実用的なインターフェースを実現するためには 1. 節で述べた (1) ~ (3) の要件が要求されるが、我々の手法は 3 要件すべてを満たすという特徴を持つ。我々の手法では要件 (2), (3) を満たすために文書画像から連結成分を取得し、連結成分ごとに認識を行う。さらに、アフィン不変量に基づく特徴量計算やハッシュ探索を行うことにより多種のフォントを登録した際にも要件 (1) の高速性を満たす。

連結成分ごとの認識では 'N', 'Z', 'z' など、互いにアフィン変換の関係にある文字を識別することができないため、そのような文字群は図 2 のように同クラスとみなし、認識時にはクラスに属する複数の文字を結果候補として出力する。文字単位認識のみではクラスレベルまでの識別しか行えないが、次節で述べる単語認識の段階で 1 字種レベルでの識別を行う。しか

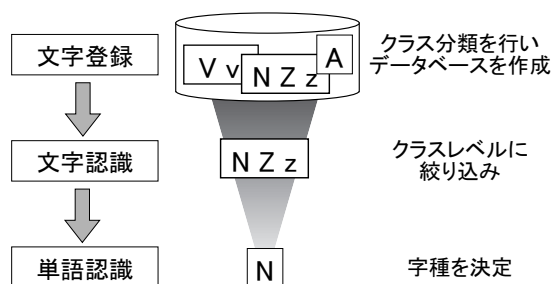


図 2 字種決定までの流れ.

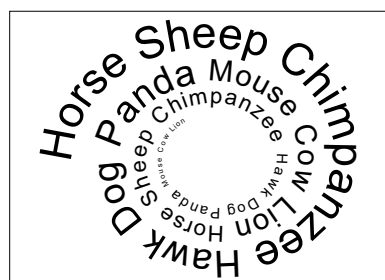


図 3 文字行を成さない文書.

し、アフィン変換の関係にある文字の組み合わせはフォントによってまちまちであるため、登録させるフォントの種類が増えるにつれ、手動でのクラス分類は困難になる。そこで、我々の手法では文字の登録時に自動的にクラス分類を行う。文字画像がデータベースに新たに追加される時、既にデータベース中に存在する文字画像とのマッチングを行い、類似した文字画像が見つかった場合には新たに登録された文字をその類似文字のクラスに分類するという方法である。

以上のように作成したデータベースを用いて認識を行うと連結成分をクラスレベルで識別することができる。このとき同時に文書画像の紙面および文字の姿勢がアフィン変換パラメータとして得られる。アフィン変換パラメータは独立変倍、シアー、回転、スケーリングに分けて推定される。全ての文字が同一平面の紙面に書かれていると仮定すると、独立変倍とシアーは紙面上の全ての連結成分において共通であり、これを利用して紙面の歪みを取り除くことができる。回転は各文字ごとに得られ、回転は単語認識処理に利用される。スケーリングも文字ごとに得られるパラメータであるが、本手法では単語認識には利用されない。なお、'i', 'j' のような複数の連結成分で構成される文字は登録時に連結成分同士のクラスや位置関係などを記録し、認識時にクラスを識別した後に統合して 1 つの文字に復元する。

ここまで書いた処理により 1 文字ごとのクラスおよびその姿勢を知ることが出来たが、どの文字の姿勢が尤もらしいか推定し、複数の文字を含むクラスから最終的な結果 1 文字を決定する処理は次節の単語認識の段階で行う。

4. 単語認識

4.1 問題設定

先に書いた通り、我々の手法では 1 文字ごとに認識を行うため、文書のどこにどの文字が書かれているかを知ることは出来るがどんな意味の単語や文章が書かれているかを知ることは出来ない。情報検索のキーワードは多くの場合単語であるということ考えると単語認識は不可欠である。

今回は英語のように単語間がスペースで区切られ、かつ左から右に記される言語に対処した単語認識を行う。また、前提条件として文書画像は文字認識の段階で独立変倍とシアーの歪みを取り除かれたものとする。我々は文字の向きを利用して文書中の文字を連結し、尤もらしい単語を求める手法を提案する。「文字の向き」とは 3. 節の文字認識で求められる回転のパラメータを指す。文字行を利用して文字の並びを推定する手法 [6]

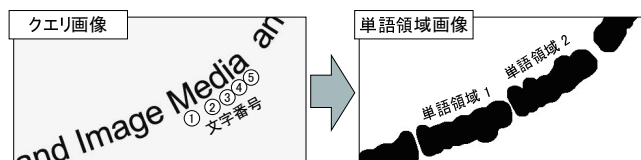


図 4 単語領域の取得.

では文字行が平行な直線である文書に限って認識可能であるが、提案手法では図 3 のような文字行を成さない文書にも対処できる。

4.2 手法

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 4 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。適切なぼかしの度合いはキャプチャ画像中の文字の間隔と太さによって変化するため、提案手法では文字間の距離および各文字の面積を計算し、それに比例したぼかしの度合いを逐次的に決定する。文字間の距離とはキャプチャ画像中の各文字から最近傍の位置にある文字とのユークリッド距離を求め、平均したものである。面積とは連結成分の画素数であり、これも平均値を用いる。ぼかしはガウシアンフィルタを用い、平均文字間距離を d 、平均面積を a とすると、ガウシアン標準偏差 σ は $\sigma = 200 d/a$ とした。また、ぼかした画像の二値化処理には OpenCV の適応二値化を用いた。

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 4 を例に挙げると、単語領域 2 から抽出される文字は ① ~ 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であるとする。

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。提案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。

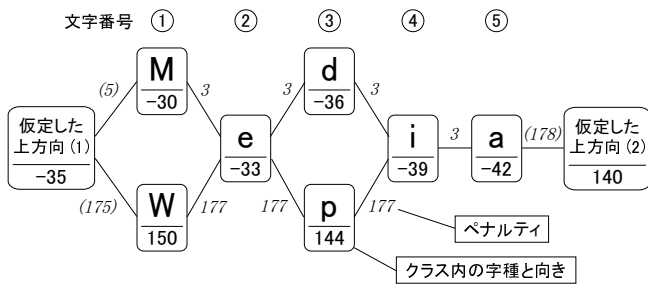


図5 文字の探索とペナルティの計算.

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4の単語領域2の単語を“④⑤①②③”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。提案手法では尤もらしい並びを求めめる方法として、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“①②③④⑤”およびその逆順“⑤④③②①”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単語の候補を挙げる。図5は推定された並び順に単語領域2の各文字の候補を挙げたグラフである。図中の数値および「仮定した上方向」は後述する。文字①から順に、また逆に⑤からも順にグラフを辿って字種を組み合わせると、“Media”、“Wepia”、“aideM”などが挙げられる。もし、単語領域2の単語を“Wepia”と読もうとすると‘W’と‘p’が他の文字と比べて約180度回転することになり、先述の仮定を考えると適した配置とは言えない。また、文字の方向が揃っていたとしても“aideM”は右から左へ読むことになるため、これも適さない。そこで提案手法では文字の向きに関するペナルティを2種類設けて尤もらしい単語候補を求めめる。ペナルティの計算には文字認識段階で得られる文字の回転角を利用する。回転角はキャプチャ画面の上方向を0度として右回転を正とする。図5の各字種の下の数値は向きを示す。

1つ目のペナルティは「文字の向きは急激に変化しない」という仮定に基づき、文字のノードを辿る際に加算される。1つ前の文字との回転角の差が小さいほど文字の向きが揃っていると言えるのでこの角度差をペナルティとする。角度差の定義域は0°～180°である。例えば図5において‘e’、‘d’と進んだ時、2つの文字の向きは3°違うため、ペナルティとして3を加算する。経路の中には何度も大きなペナルティが加算されるような組み合わせが存在するが、その場合は途中で計算を打ち切り、候補から除外することで処理時間の増加を抑える。

2つ目のペナルティは「単語は左から右に読む」というルールに関するものである。単語領域2の単語を例に取り、考え方を図6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6(a)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、この差の値をペ



(a) 文字番号①から読む場合 (b) 文字番号⑤から読む場合

図6 上方向の仮定.

ナルティとする。図5において「仮定した上方向(1)」および次ノードとの間のペナルティが計算されることになり、上方向が -35° だったとすると文字①が‘M’の場合ペナルティは5, ‘W’の場合は175である。“aideM”のように単語を⑤から順に読むと仮定した時は図6(b)のように上方向が仮定され、図5における「仮定した上方向(2)」が求まることになる。

以上のように文字候補の組み合わせごとにペナルティの合計を求めた後、昇順にソートすることで尤もらしい文書中の単語が推定できる。例で用いた単語領域2の場合、“Media”がペナルティ17で最小であった。1文字ごとの認識では区別できなかった‘d’と‘p’のような同一クラスの字種も単語認識の段階で文字レベルで区別されるようになった。しかし、‘0’と‘O’のように互いに拡大縮小の関係にあり、向きが類似する字種についてはペナルティがほぼ等しくなり、尤もらしい字種の決定が困難である。我々の対処方法はペナルティの小さい複数の候補を全て情報検索システムのキーワードとして使い、リンクが貼られたものがあれば、それが尤もらしい単語であると見なし、字種の決定やリンク先へのアクセスを行うというものである。また、将来的には誤認識の文字が含まれる単語でもあいまい検索によりアクセスが出来る機能を検討している。

2.2節でフォント別の情報提供を行う機能を紹介したが、提案手法では簡易的な方法で単語単位のフォント推定を行う。ペナルティの計算をしながら文字を探索する際、各ノードに字種と回転角の他にフォント情報も保持し、1文字辿るごとにフォントのヒストグラムに投票する。そして、終端の文字までたどり着き、単語の候補が1つ生成された時点で最頻であるフォントをその単語のフォントと推定する。将来的にはフォントに関するペナルティも設け、隣接する文字とフォントが同じであるかどうか考慮しながら推定を行い、フォント識別の精度を向上させる必要がある。

5. 実 験

提案手法の有効性を確認するため、カメラで撮影した文書の単語を認識する実験を行った。用いた計算機はCPUがOpteron 2.8GHz、メモリが16GBである。実験ではデータベースに登録したフォントの種類が増えた時に認識精度や処理時間がどのように変化するかを調べた。

フォントは図7に示す10種類を対象とし、1～10種類のフォントを登録したデータベースをそれぞれ作成し、認識を行った。1種類の場合はArialのみ、2種類の場合はArialとCentury、のように上から順にデータベースの作成と認識に用いた。各フォントで登録される字種はアルファベットの太文字

Arial 0123ABCDabcd
 Century 0123ABCDabcd
 Times New Roman 0123ABCDabcd
 Verdana 0123ABCDabcd
Meiryo bold 0123ABCDabcd
 OCRB 0123ABCDabcd
Book Antiqua Bold 0123ABCDabcd
BellGothicStd-Black 0123ABCDabcd
Franklin Gothic Medium 0123ABCDabcd
 TektonPro-Bold 0123ABCDabcd

図 7 実験に用いたフォント。

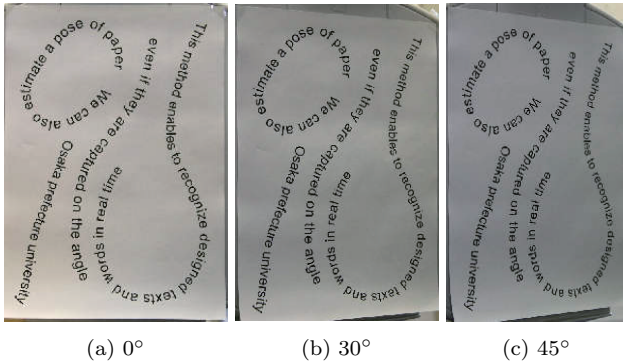


図 8 認識対象 (Arial)。

と小文字, および数字の計 62 種である。

文字登録時にはカメラ撮影時のピンぼけや解像度の低下に対処するため, 生成型学習法 [7] を用いる。本実験ではガウスぼかしを 3 段階, 解像度低下を 3 段階 (ただし, ぼかし無し・解像度変化無しの段階も含む) の計 9 段階の劣化を適用した。そのため, 10 フォントでは文字画像 5,580 枚分のデータが登録されることになる。

認識対象は図 8 に示す文書であり, 144 文字, 30 単語の英文が曲線上に並べて書かれている。ただし, 文書に 1 つ含まれる冠詞の “a” は単語認識処理を行う必要が無いため, 単語数には含んでいない。この文書をデータベースと同じフォントで計 10 枚, A4 用紙に印刷した。この用紙を正面に対して 0 度, 30 度, 45 度の角度をつけてカメラで撮影した。図 8 (a) ~ (c) に Arial の場合の撮影画像を示す。Arial の文書を 0 度から撮影した場合, 画像サイズが 1633 × 2333 ピクセル, 1 文字あたりの平均サイズが 58.2 × 48.0 ピクセルであった。

文字ごとにクラスレベルの認識を行った後, 単語の候補を挙げて 4.2 節で述べたペナルティを昇順に並べた。そして, 各単語について最小ペナルティの単語 1 個のみ, または上位 10 個までの単語を見たときに正解の単語が含まれるかどうかを調べて単語認識率を求めた。なお, 先述の通り本手法では大文字と小文字が拡大縮小の関係にある字種を区別することが困難なため, 単語中の ‘C’, ‘O’, ‘S’, ‘V’, ‘W’, ‘X’, ‘Z’ については大文字と小文字の違いがあっても正解とした。フォントの推定は性能評価の対象とせず, 字種ごとの比較とした。なお, 予備実験でフォントも比較したところ, 0 度から撮影した文書 10 フォント分の単語認識率は最小ペナルティのみを見た場合で 60.0 %,

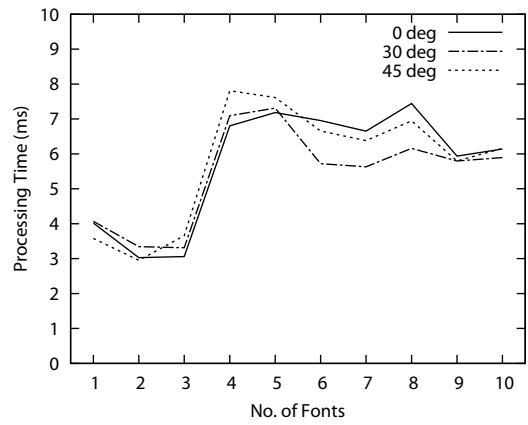


図 9 1 単語あたりの単語認識処理時間。

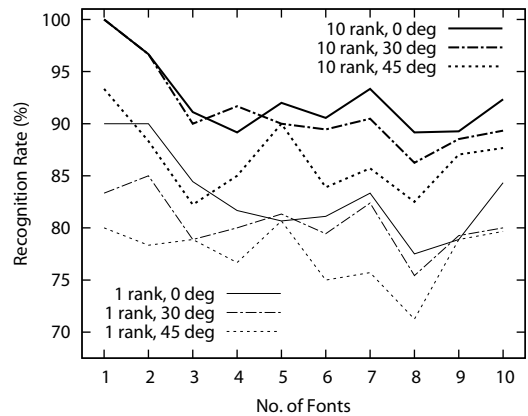


図 10 単語認識率。

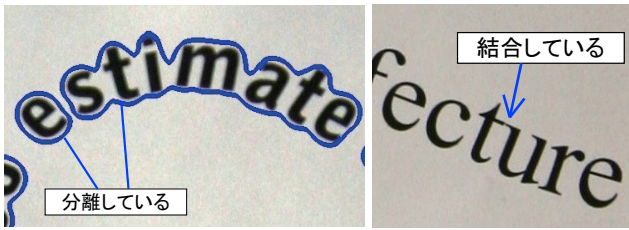
上位 10 単語を見た場合で 72.0 % であった。

図 9 に 1 単語あたりの処理時間を示す。処理時間は単語認識に要した時間であり, 文字認識は含んでいない。データベースに登録したフォント数によって処理時間に大きなばらつきがあり, フォントの種類によって処理時間はまちまちであることが分かった。10 フォント, 0 度の場合, 処理時間は 6.14 ミリ秒であった。処理時間増加の要因としては登録画像の増加によって 1 クラスあたりの字種の数が増え, ペナルティの計算量が増加したためと考えられる。表 1 に Arial を登録した時点で同一クラスとなった文字群を示す。

次に, 単語認識の結果を図 10 に示す。グラフにはペナルティの小さい順から 1 位の単語のみを見たときの正解率と 10 位までの単語を見たときの正解率が撮影角度別で描かれている。概ね扱うフォント数の増加に伴い認識率が低下したが, 10 位までの単語を調べることで 1 位のみ調べるよりも平均 9.7 % 上昇し, 10 フォント, 0 度で 92.3 % の認識率が得られた。1 位のみで正解の単語がカバーできない原因は同クラスで向きが類似した不正解文字の方がより小さいペナルティを得たためであ

表 1 文字認識において同一クラスに統合された字種 (Arial) . 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	



(a) 単語領域の取得失敗 (b) 隣接文字との結合

図 11 単語認識・文字認識の失敗例.

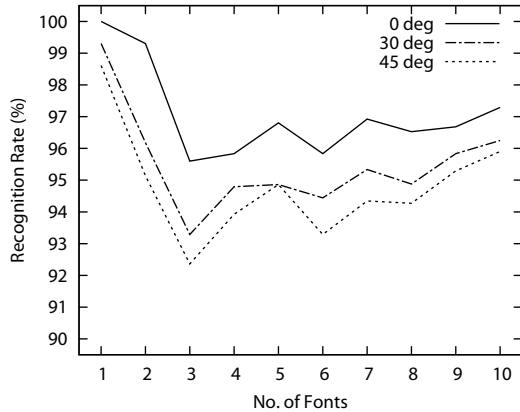


図 12 文字認識率.

る．なお，20 位までを調べた場合は 10 位の場合より平均 0.4 % の上昇に留まったため，これ以上はペナルティの逆転以外の要因による認識失敗である．単語認識の失敗例としては文字単位でのクラス絞り込みに失敗したことや単語領域の取得に失敗したことが挙げられる．単語領域の取得失敗の例を図 11 (a) に示す．文字列を囲む枠線は推定した単語領域の輪郭を表し，“estimate” の ‘e’ とそれ以外の文字が過分割されてしまっている．この単語を認識できるように画像のぼかしを強くすると別の文書画像においてスペースを隔てて複数の単語が結合してしまうことを確認した．そのため，ぼかしの強さの決定法を変えること，もしくは画像をぼかす方法以外で単語領域を取得することが今後の課題となる．

図 12 は検証のために文字単位での認識率を求めたものである．クラスレベルでの認識を行い，得られたクラスの中に正解の字種が含まれていた場合に認識成功と見なした．文字認識失敗の例としてはデータベースのデータ数が増加したことによる誤認識や図 11 (b) の ‘t’ と ‘u’ のように隣接した文字と連結成分が統合されて文字単位の認識が不可能となったことが挙げられる．文字単位の認識が 1 つ失敗すると，たとえ他の文字の認識が成功していても 1 単語の認識が不可能になってしまうため，単語認識の精度には文字単位での認識精度が非常に大きく関わると言える．単語認識率向上のためには文字認識技術の改良が重要である．

最後にアプリケーションの利便性に関わるその他の性能について述べる．単語認識処理に要する時間は図 9 に示した通りであるが，文字認識処理に要する時間は 10 フォント，0 度の場合で 1 文字あたり 3.44 ミリ秒であった．5 文字で 1 単語の文書を認識させた場合，文字認識 5 回分と単語認識を合わせて

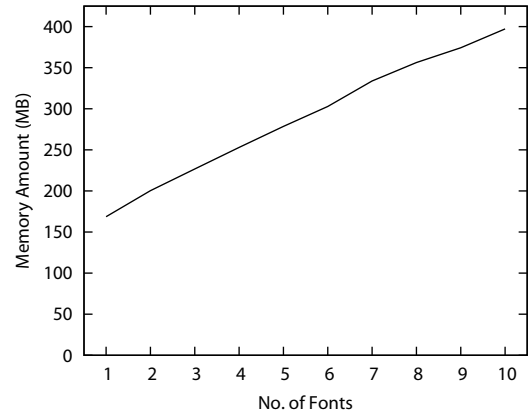


図 13 メモリ使用量.

23.34 ミリ秒となり，1 秒間に約 42 単語を処理できることになる．これにより市販のノートパソコンとカメラでも実時間で動作するインタフェースが実装可能と言える．図 13 はデータベースを読み込んだ時のメモリ使用量である．10 フォントの英数字を扱うためには約 397 MB を要した．今後さらに多くのフォントや他言語の文字を扱うためにはメモリ削減が重要な課題となる．

6. おわりに

本稿では実時間文字認識技術を活用したカメラベース情報取得システムを提案した．新たな対話型インタフェースの可能性について述べ，デモとして環境中文字列をリンクアンカー化するアプリケーションを紹介した．また，利便性を考慮した単語認識手法を提案し，実験によって有効性を示した．今後の課題としては文字領域および単語領域の取得方法を検討すること，より多くの文字に対応するためにメモリを削減すること，そしてフォント認識の検討および性能評価が挙げられる．

謝辞 本研究の一部は，科研費補助金若手研究 (B) 21700202 ならびに平成 21 年度シーズ発掘試験 (発掘型) (課題番号 11-084) の補助による．

文 献

- [1] 辻智彦, 堀松晃, 岩村雅一, 黄瀬浩一, “文字の並びをリンクアンカー化する web カメラベースインタフェース,” MIRU2009 論文集, pp.1863–1864, July 2009.
- [2] 岩村雅一, 辻智彦, 堀松晃, 黄瀬浩一, “レイアウト非依存な実時間カメラベース文字認識,” MIRU2009 論文集, pp.174–181, July 2009.
- [3] M. Iwamura, T. Tsuji, A. Horimatsu and K. Kise, “Real-time camera-based recognition of characters and pictograms,” Proc. ICDAR2009, pp.76–80, July 2009.
- [4] 岩村雅一, 辻智彦, 黄瀬浩一, “カメラ撮影文字の事例に基づく実時間認識,” 信学技報, Feb. 2010.
- [5] Y. Watanabe, Y. Okada, Y.-B. Kim and T. Takeda, “Translation camera,” Proc. ICPR1998, pp.613–617, 1998.
- [6] G. K. Myers, R. C. Bolles, Q.-T. Luong, J. A. Herson and H. B. Aradhye, “Rectification and recognition of text in 3-d scenes,” IJDAR, vol.7, no.2-3, pp.147–158, 2004.
- [7] H. Ishida, S. Yanadume, T. Takahashi, I. Ide, Y. Mekada and H. Murase, “Recognition of low-resolution characters by a generative learning method,” Proc. CBDAR2005, pp.45–51, Aug. 2005.