

2 ストローク入力のための仮名漢字変換†

塩 見 彰 睦** 喜 多 辰 臣**
河 合 和 久*** 大 岩 元***

日本語 2 ストローク入力との併用に適した仮名漢字変換方式として、漢字混じり仮名漢字変換を提案し、その実現について述べる。日本語 2 ストローク入力と従来の仮名漢字変換入力を併用した場合、入力したい熟語を構成する漢字のうち、一つでもその漢字の 2 ストロークコードを入力できないときには、すべての漢字をその読みで入力し、変換しなければならなかった。この欠点を解決するのが、漢字と読みが混じった文字列を熟語に変換する漢字混じり仮名漢字変換である。本論文では、漢字混じり仮名漢字変換を行うための方式として、変換用辞書の見出し語を漢字混じり語に拡張するものと、入力文字列に含まれる漢字をキーに辞書引きを行う漢和辞書を用いるものとを提案する。さらに、この二つの方式でパーソナル・コンピュータ上に漢字混じり仮名漢字変換システムを実現し、その比較を行った。その結果、変換速度ならびにパーソナル・コンピュータ環境での実現性の点で、漢和辞書を用いる方式が優れていることが明らかになった。また、同方式は MS-DOS 上のデバイス・ドライバとして実現され、いくつかのエディタやデータベース、日本語ワープロなどの市販ソフトウェアに試用され、実用に供しうる日本語入力システムであることが確認された。

1. はじめに

2 ストローク入力方式¹⁾の一つである TUT コード²⁾は、仮名漢字変換を利用せずに漢字コードと呼ばれる 2~3 打鍵で漢字を直接入力できる。このため、日本語の高速入力が可能となり、主として出版業界などで、日本語入力を業務とする専任コピータイピストに広く利用されている。また、コピータイプのみならず創作者の日本語入力としても有効であることが報告されている³⁾。

専任タイピストの場合は、業務に入る前に長い練習時間をかけて、多くの漢字コードを覚えている。しかし、山田の指摘⁴⁾にもあるように、一般ユーザにとっては漢字コードを覚えるのは難しいし、難しいという気持ちが覚えることの障壁となっている。

そこで、TUT コードでは、仮名を行段構造に従って系統的にコード化し、短時間で覚えらるるようになっている。これによって、一般ユーザでも、仮名コード

と仮名漢字変換を用いて、容易に日本語入力を実用化できる。彼らは、仮名漢字変換を用いながら、自分がよく使う漢字の漢字コードを徐々に覚えていき、覚えた漢字を直接入力することによって、日本語入力の高速化を進めていくわけである。

ところで、従来の仮名漢字変換を TUT コードと併用する場合、入力したい熟語がいくつかの漢字からなり、そのうちの一つでもその漢字コードを覚えていないときには、漢字コードを覚えている漢字も含めて、すべての漢字をその読み(仮名)で入力し、変換しなければならない。言いかえると、漢字コードを覚えるならば、熟語単位で覚えないと有効に使えないということである。これでは、せっかく覚えた漢字コードを有効に活用できず、ひいては、漢字コードを覚える意欲が減退してしまい、漢字コードを覚えて、漢字を高速に入力するという TUT コードの長所が活かせない。そこで、われわれは TUT コードのような無想式 2 ストローク入力方式との併用に適した仮名漢字変換方式として、漢字混じり仮名漢字変換を提案した^{6),7)}。

本研究と同種の研究に、山田のグループによる T コード用仮名漢字変換方式「交ぜ書き変換」^{4),5)}がある。「交ぜ書き変換」では、辞書の大きさを小さくするためにコードの修得グレード別の辞書を用いている。

2 章では、2 ストローク入力方式との併用に適した漢字混じり仮名漢字変換について述べ、3 章では漢字混じり仮名漢字変換を交ぜ書き変換方式と漢和辞書方式の二つの方式で実現し、評価を行ったことについて述べる。4 章では、実用システムとして実現した MS-

† A Kana to Kanji Conversion for Two-Stroke Input Method of Japanese Text by AKICHIKA SHIOMI, TATSUOMI KITA (Department of Information and Computer Sciences, Toyohashi University of Technology), KAZUHISA KAWAI (Department of Knowledge-based Information Engineering, Toyohashi University of Technology) and HAJIME OHIWA (Department of Information and Computer Sciences, Toyohashi University of Technology).

†† 豊橋技術科学大学情報工学系
††† 豊橋技術科学大学知識情報工学系

* 現在 富士ゼロックス(株)
Present Affiliation Fuji-XEROX Corp.

** 現在 慶應義塾大学環境情報学部
Present Affiliation Faculty of Environmental Information,
Keio University

DOS のデバイス・ドライバについて述べる。

2. 漢字混じり仮名漢字変換

漢字混じり仮名漢字変換は、漢字混じり仮名文字列を熟語に変換するものである。すなわち、ある熟語を変換入力するとき、その熟語を構成する漢字のいくつかを漢字コードで直接入力できるならば、それらの漢字を漢字コードで入力し、他の部分を熟語の読み、つまり、仮名で入力する。こうして入力された漢字混じり仮名文字列に対して変換を行い、熟語を得ることである(図1参照)。なお、本研究では、変換方式として、いわゆる(単)文節変換を用いている。

漢字混じり仮名漢字変換を用いることにより、従来の仮名漢字変換を用いる場合に比べ、一度覚えた漢字コードを熟語入力に利用できる率が向上する。TUTコードのように連想のない2ストローク入力方式では、漢字コードは、無意識下の手指の運動として習得されるものであり、いくども漢字コードを打っている間に自然と身についていくものである。したがって、漢字の利用率が向上することは、漢字コード習得に効果的である。

図2は、新聞に用いられている漢字を含む熟語⁹⁾約28万語と、熟語に含まれる漢字をもとに、覚えた漢字コードで直接入力できる熟語数と、直接入力と漢字混じり仮名漢字変換を併用して入力できる熟語数を漢字コードの習得数に対してグラフ化したものである。ここで、漢字コードは頻度の高いものから順に覚えていったものとしている。この2本のグラフの差が漢字混じり仮名漢字変換を用いて入力できる語数である。覚えている漢字コード数の少ない時期ほどその差が大きく、漢字混じり仮名漢字変換が漢字コードの利用率向上に有効であることが図から読み取れる。仮名漢字変換を併用しながら、漢字コードを徐々に覚えていくTUTコードの一般ユーザ、特に習得漢字コード数の少ないユーザにとっては、漢字混じり仮名漢字変換が有効といえる。

漢字混じり仮名漢字変換では、従来の仮名漢字変換に比べ、入力文字列に含まれる漢字の持つ情報を利用できることから、変換の性

能が向上することが期待できる。ここでは、変換の性能を示す指標として、

(1) 変換速度：(再)変換キーを押してから、(次)候補が得られるまでの時間。

(2) ヒット率：入力したい熟語が、第一候補として得られる率。

の二つをとりあげる。このうち、変換速度は、変換に用いるアルゴリズムならびに辞書の構造に大きく依存するので、次章で、漢字混じり仮名漢字変換の実現方式ごとに検討する。

ヒット率は、基本的には、変換に用いられる辞書に含まれる同音異義語の数で決定される。表1は、われわれが使用している辞書元であるVJE-αの仮名漢字変換用辞書の見出し語と、その熟語、および、熟語を構成する漢字をもとに⁹⁾、従来の仮名漢字変換と、漢字混じり仮名漢字変換における同音異義語に関する指標を算出したものである。表中のいずれの数値から、漢字混じり仮名漢字変換を用いることにより、ヒット率が向上することが推察される。特に、同音異義語を持つ見出し語の割合が半減していることは、変換

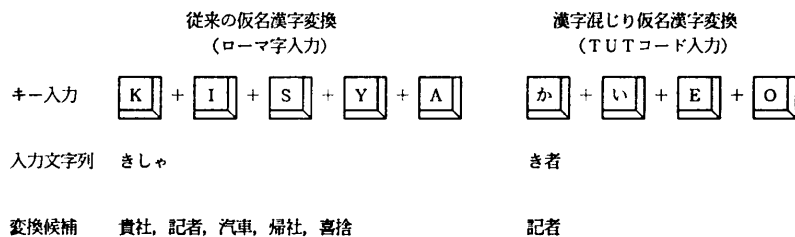


図1 漢字混じり仮名漢字変換

Fig. 1 A kana to kanji conversion for mixed input of kana and kanji combination.

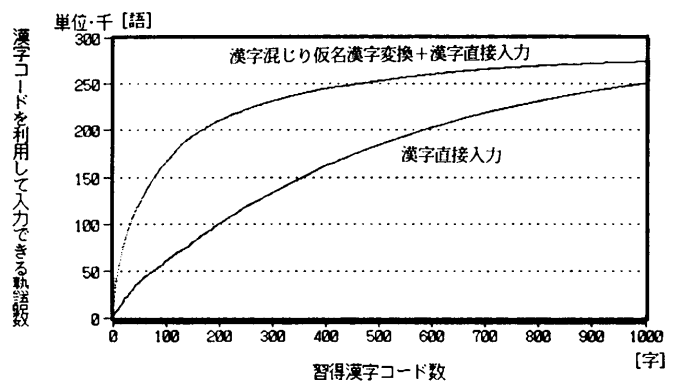


図2 漢字コード利用率の比較

Fig. 2 Number of inputtable jyukugo vs. learned kanji code for direct input and mixed input kana kanji conversion.

の結果、ただ一つの熟語候補が得られる場合が増えることを意味し、頻度情報などを用いた変換候補の提示順決定アルゴリズムの優劣に関係なく、ヒット率が向上することを示している。

一方、漢字混じり仮名漢字変換の欠点としては、従来の仮名漢字変換に比べ、変換アルゴリズムが複雑になる、あるいは、辞書が大きくなるなどが考えられる。これらは、変換速度と同様に、実現方式に依存するため、次章で検討する。

3. 漢字混じり仮名漢字変換の実現と評価

本章では、パーソナル・コンピュータ（以下、パソコンとする）上に試作した二つの漢字混じり仮名漢字変換方式—交ぜ書き辞書方式と漢和辞書方式—の変換アルゴリズムと辞書構成について述べる。さらに、両方式の性能を比較、検討し、その実用性を評価する。

3.1 交ぜ書き辞書方式

漢字混じり仮名漢字変換法として、まず考えられるのが、入力される漢字混じり仮名文字列をそのまま変換用辞書の見出し語とする交ぜ書き辞書方式である。従来の仮名漢字変換用辞書の見出し語が、読みを表す仮名のみからなる文字列であったのを、漢字混じり仮名文字列に拡張するわけである（図3参照）。

小野の「交ぜ書き変換」法に用いられている変換用辞書はこの方式である。小野は、合成語を分解することと、見出し語に使用する漢字を「習得グレード」別を選択することで、辞書の膨張を5割増程度に抑えている⁵⁾。

交ぜ書き方式では、変換される対象文字列が仮名文字列から漢字混じり仮名文字列に拡張されるだけで、変換アルゴリズムそのものは、従来の仮名漢字変換のそれをそのまま適用できる。

辞書は、漢字混じり仮名文字列である見出し語とそ

の熟語および品詞情報が、見出し語のコード順に配列されたものである。同音異義語の場合、一つの見出し語に、いくつかの熟語とその品詞情報の組が格納されている。以下では、見出し語とその熟語、品詞情報の組を辞書項目と呼ぶ。

本研究では、変換方式として、文節変換を用いているので、基本的に、変換処理は辞書引きと付属語処理の繰り返しとなる。変換処理（図4参照）は、まず、入力文字列によって辞書引きを行うことから始まる。辞書中に入力文字列を見出し語とする辞書項目があれば、その熟語を変換結果とする。辞書項目がなければ、入力文字列の後方から1文字取り除いた文字列で同様に辞書引きを行い、熟語が得られれば、その品詞情報をもとに、先ほど取り除いた文字列が、付属語としてその熟語に接続するかを調べる。文法的に正しい付属語と判定されれば、変換結果として、熟語と取り除いた文字列を返す。辞書引きの結果、辞書項目がないか、あるいは、取り除いた文字列が付属語として不

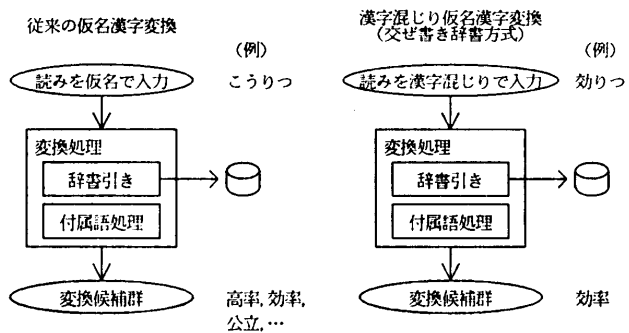


図3 交ぜ書き辞書方式
Fig. 3 Mixed form conversion method.

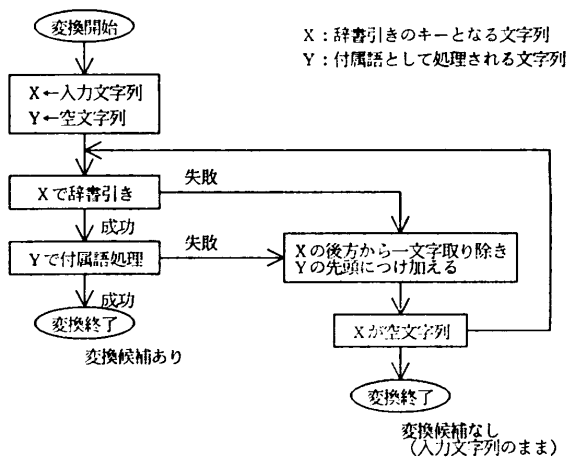


図4 変換アルゴリズム
Fig. 4 Conversion algorithm.

表1 同音異義語に関する指標
Table 1 Statistics of homophones.

指 標	従 来 の 仮名漢字変換	漢 字 混 じ り 仮名漢字変換
全見出し語の持つ同音異義語数の平均値	1.415 語	1.152 語
同音異義語を持つ見出し語の割合	19.627%	9.446%
同音異義語を持つ見出し語のみを対象とした同音異義語数の平均値	3.088 語	2.615 語

適格であった場合は、入力文字列からさらに1文字取り除いて、同様の処理を繰り返す。最終的に、入力文字列がすべて取り除かれてしまったときは、入力文字列そのものを変換結果とする。以上が、変換処理の基本的なアルゴリズムである。これより、変換速度は辞書引きの速度に大きく依存することは明らかである。

本方式では、図5のように、辞書引きの高速化のために、2段階のインデクシングを行っている。辞書は、大きさ2KBのレコードに細分されており、各レコードの先頭の見出し語が、インデクスに順に置かれている。インデクスはメモリ上に常駐され、入力された漢字混じり仮名文字列が、どのレコード内にあるか、このインデクスから求められ、そのレコードをメモリ上に転送する。レコードは、サブインデクスとレコード本体からなり、サブインデクスと入力文字列を比較し、一致したらレコード本体から目的とする辞書項目を得る。

また、図5に見られるように、見出し語は2バイト（シフトJIS）コードで表現されている。これは、見出し語が漢字混じり仮名文字列であるためである。

3.2 漢和辞書方式

市販の漢和辞書には、ある漢字が使われている熟語とその意味が、漢字ごとにまとめられ、漢字の部首順に並べられている。そこでわれわれは、ある漢字を含んだ漢字混じり仮名文字列とその熟語と品詞情報を、漢字ごとにまとめて漢和辞書とし、これを用いて、漢字混じり仮名漢字変換を実現した。

変換は、図6のように、入力文字列に漢字が含まれるかを調べることから始まる。漢字が含まれない場合には、従来の仮名漢字変換を行う。漢字が含まれているときには、漢和辞書でその漢字の部分を検索し、入力文字列と一致するものを探し、その熟語を変換結果とする。従来の仮名漢字変換も、漢和辞書を用いた変換も、前節で述べた辞書引きと付属語処理からなる変

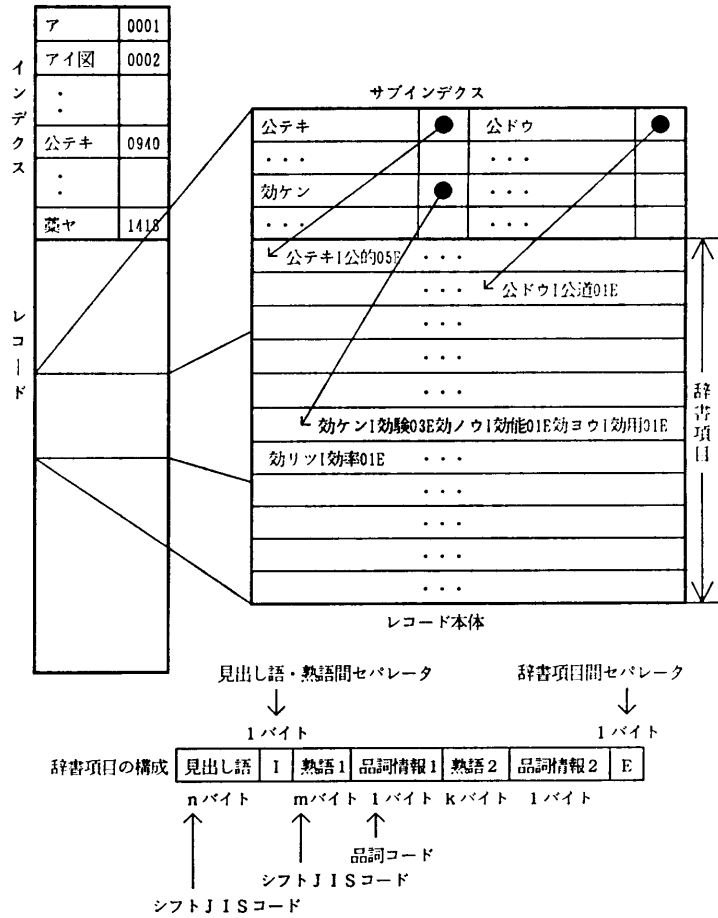


図5 交ぜ書き辞書方式の辞書の構造
Fig. 5 Schematic diagram of the dictionary used in the mixed form conversion.

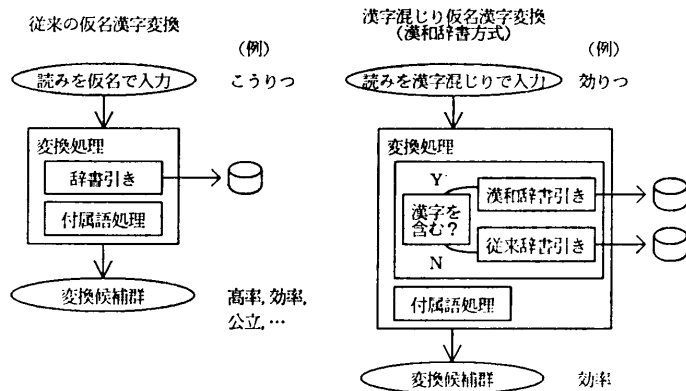


図6 漢和辞書方式
Fig. 6 Kanji lexicon method for mixed form conversion.

換アルゴリズムを用いている。特に、付属語処理は、交ぜ書き辞書方式と漢和辞書方式の2種の漢字混じり仮名漢字変換と、比較のための従来の仮名漢字変換すべてに共通のものである。

漢和辞書は、図7のように、インデクスとレコードからなる。一つのレコードには、特定の漢字が見出し語に用いられている熟語をすべて格納してある。したがって、交ぜ書き辞書とは異なり、レコードの大きさは一定でない。インデクスには、漢字に対応するレコードを指すポインタが、漢字のコード順に用意されており、あらかじめメモリ上に常駐される。

検索の実行は、インデクス上の、入力文字列に含まれる漢字のコード位置にあるポインタ値により、レコードを得て、メモリ上に転送し、さらにレコード内から目的とする熟語を得ることによって達成される。レコードには、交ぜ書き辞書と同様に、サブインデクスが付けられている。

漢和辞書は、交ぜ書き辞書とは異なり、見出し語を1バイトコードで表現し、辞書の縮小を図っている。これは、見出し語に現れるのが、検索キーである漢字であることを表す1バイトコード(図7中ではXと表示。以下、このコードをXとする)と、仮名文字だけのためである。漢和辞書に対する辞書引きにおいては、文字列中の漢字をXとし、仮名の部分と合わせ、一致する見出し語を探すわけである。

ところが、「衣食住」のような三つの漢字からなる熟語を入力する際に、「衣」と「住」を漢字で、「しょく」を仮名で入力する場合がある。本方式では、こういう場合のために、文字列の最初に現れる漢字をXとし残りの漢字を別のコード(Y)で表す見出し語を用意している。一致する見出し語の検索処理においては、「衣しょく住」の入力に対し、「衣」のレコードから「XしょくY」と一致する見出し語を検索し、見出し語のYの部分の漢字「住」と辞書項目の熟語にある対応する漢字「住」が一致したものを変換候補とする。

漢和辞書を作成するにあたり、以下に挙げる点が問題となった。

- (1) 見出し語を分割できない場合
例) 今日, 大和
- (2) 一部見出し語が分割できない場合
例) 今日子, 大和絵

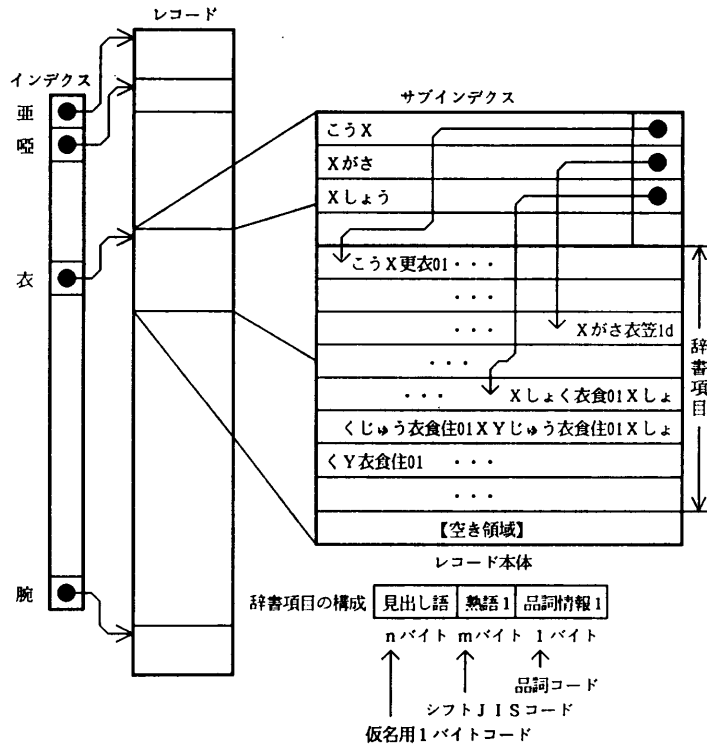


図7 漢和辞書方式の辞書の構造
Fig. 7 Schematic diagram of the kanji dictionary used in mixed form conversion.

(3) 熟語中の文字が省略されている場合

例) 一宮 (「一の宮」の「の」が省略されている)
まず、(1)の場合は入力の際に漢字と仮名の部分を分割することができないため、この熟語が漢字混じり文字列として入力されることはないと考えた。すなわち、必ず仮名のみか漢字のみで入力されるものとしたわけである。このため、この問題に対しては、特に対策を施す必要はない。

これに対して、(2)の場合は、「今日こ」と入力することが考えられる。この入力に対しては、上述した「衣食住」の場合と同様の方法で処理できる。すなわち、辞書の「今」のレコード中に、見出し語が「XYこ」であり、熟語が「今日子」である辞書項目を登録しておき、入力文字列の漢字「日」と熟語中の漢字「日」の一致を調べるわけである。

最後に、(3)の場合において、漢字混じりの入力文字列として考えられるのは「一のみや」、「一みや」、「いちの宮」、「いち宮」の4通りである。これらの入力すべてにおいて、「一宮」を検索するため、「一」の見出し語に、「Xのみや」、「Xみや」、「宮」の見出し語には、「いちのX」、「いちX」を登録した。すなわ

ち、それぞれの漢字の見出し語に「の」を含むものと含まないものの2通りを用意するわけである。

なお、これらの問題は、交ぜ書き辞書方式においても同様に生じるが、同様の手法で解決される。

3.3 実現方式の比較・評価

上記二方式を比較する。評価は、それぞれの方法に必要な辞書の大きさと、変換システムの大きさ、および変換速度をもって行う。

パソコンが普及した今日、日本語入力方式の評価を行う場合も、パソコン上での実用性を考慮しなければならない。パソコンの限られたディスク容量では、辞書の大きさは、性能比較の重要な一項目となる。

また、パソコンの日本語入力は、多くの場合、デバイス・ドライバとして作製される。デバイス・ドライバが、限られたメモリ容量の多くを使用してしまえば、日本語ワード・プロセッサなどのアプリケーション・システムの動作速度が低下したり、メモリ不足で動作しなくなってしまう。逆の言い方をすれば、市販の日本語入力用デバイス・ドライバ程度の大きさの変換システムで、いかに高速な変換処理を実現するかが重要ということである。したがって、変換システムの大きさも重要な評価基準となる。

しかし、最近では増設 RAM を RAM ディスクのような高速外部記憶装置としてではなく、内部拡張メモリとして用いる方法 (EMS 方式)¹⁰⁾ が開発されている。現在では、EMS を利用して占有するメモリ領域を少なくするデバイス・ドライバもある。この場合、デバイス・ドライバもより大きなものにすることができる。

表 2 は、辞書と変換システムの大きさをまとめたものである。二つの漢字混じり仮名漢字変換方式のほか、に、対照データとして、従来の仮名漢字変換ならびに

代表的な市販の日本語入力用デバイス・ドライバに關するデータを挙げている。われわれの辞書は、見出し語数約 3 万 3 千語、総熟語数約 4 万 7 千語 (単漢字を含む) の VJE- α の辞書から作成したものである。

われわれは、小野のように合成語の分解などは行わず、TUT コードで入力できる 2,525 文字のすべての漢字に対してそれぞれの辞書の登録を行った。この方式では、見出し語が 2 バイトコードであり、さらに n 文字の漢字を含む熟語の展開数 (見出し語数) は 2^{n-1} となるため、従来の仮名漢字変換辞書の 5.5 倍になっている。小野は合成語の分解を行うことで、全語を制限なしに展開しても辞書の膨張を 2.3 倍に抑えている⁵⁾。

この結果より、われわれの交ぜ書き変換方式は、現行のパソコン環境においては、フロッピー・ディスクでの利用は不可能であることがわかる。しかし、安価になったハードディスクや、大容量化された RAM ディスクを用いれば、実用に供しうるといえる。

一方、漢和辞書方式は、従来用辞書と漢和辞書を 2 枚のフロッピー・ディスクにおくことで利用できる。また、ハード・ディスク、RAM ディスクでの利用も当然可能である。変換システムの大きさは、いずれの方式においても、市販デバイス・ドライバと大差のない大きさに納まっている。

変換速度の測定実験は、以下の手順で行った。

(1) 単文節変換で辞書引きが 1 回から 4 回で変換される読みをそれぞれ 20 件用意する。辞書引きの回数は、3.1 節に述べた変換アルゴリズムからわかるように、付属語の長さで決まる。辞書引き 1 回は、付属語のないもの、辞書引き n 回は、 $n-1$ 文字の付属語 (副助詞「は」、接続助詞「のに」や助動詞「させる」など) が付いている場合である。

(2) それぞれ 20 件の読みに対して、変換キーが押されてから、変換の結果、一つ目の変換候補が得られるまで操作を 50 回繰り返し変換時間の総和を求めらる。

(3) 変換時間の総和を 1,000 で割り 1 回の変換に要する平均時間とする。

変換速度の実験結果は、表 3 (RAM ディスクを用いた場合) ならびに表 4 (ハードディスクを用いた場合) のようにまとめられる。かつ内は、従来の仮名漢字変換に要する時間を 100 としたときの、変換時間の値である。この実験で使用した従来の仮名漢字変換システムは、3.2 節で述べた漢和辞書方式において、

表 2 変換システムの性能評価 (1)
Table 2 The performance of conversion systems (1).

	辞書の大きさ	システムの大きさ
従来の仮名漢字変換 ¹⁾	531, 456 B	122, 704 B
交ぜ書き辞書法	2, 937, 856 B	128, 096 B
漢和辞書法	531, 456 B + 966, 200 B ²⁾	127, 184 B
VJE- β	483, 328 B	106, 240 B
ATOK 6	510, 464 B	91, 648 B

注: ¹⁾ TUT コード入力による仮名漢字変換システム (漢和辞書法の仮名のみを入力に対する処理部を抽出したもの)。

²⁾ 従来の仮名漢字変換用辞書と漢和辞書の大きさ。

仮名のみの見出し語入力に対する変換を行う単文節変換システムを単体で用いたものである。

なお、市販デバイス・ドライバでは、同等の変換時間を厳密に測定できないので、本実験では測定していない。

実験結果より、交ぜ書き辞書方式では、辞書が大きいため検索に時間がかかり、変換速度が遅くなっていることがわかる。これは、システムの大きさの制限から、十分なインデクシングが行えないためである。一方、漢和辞書法は、漢字コードによるインデクシングが有効に機能し、比較的高速な変換が行われている。

以上の結果から、漢字混じり仮名漢字変換を実現する二方式について、以下のような評価がまとめられる。

(1) 辞書格納に RAM ディスクを用いれば、両方式とも、従来の辞書に比べ2倍以内におさまっており、十分実用に供しうる変換速度をもっている。

(2) ヒット率の向上を考慮すれば、2ストローク入力には漢字混じり仮名漢字変換が有効である。特に、現在のシステムでは、辞書の大きさ、変換速度の点で、漢和辞書方式の方が実用性が高い。

(3) EMS などを利用してパソコンのメモリ容量が拡張された場合、インデクシングの方法などを改良することにより、交ぜ書き辞書方式も一層の速度向上が望める。

4. 実用システムの作製

漢字混じり仮名漢字変換の両方式は、PC-9801 上に、MS-DOS のデバイス・ドライバとして実現されている。MS-DOS ユーザがデバイス・ドライバを作る場合、アセンブラ言語で記述するのが通常である。本研究では、アセンブラで書かれたプログラムから、C言語で書かれたプログラムを呼び出す手法を組み込み、変換処理を行う部分をC言語で記述した。つまり、デバイス・ヘッダの記述や、セグメントの管理などアセンブラでしか書けない部分はアセンブラで記述し、変換処理などは、作製、デバッグ、保守、拡張が容易なようにC言語で記述している。

システムの実用化に際し、市販の仮名漢字変換システムの持つ種々の機能—仮名への変換機能、辞書管理機能などを組み込んだ。また、そうした機能の他に、TUT コードのための仮名漢字変換システム固有の二つの機能を組み込んでいる。

一つは、直接/間接入力モードの設定である。間接入力モードとは、通常の仮名漢字変換と同様の入力モードで、入力された文字列は変換用のバッファにためられ、変換キーが入力された時点で、バッファの内容に対して変換処理が行われるものである。

これに対し、直接入力モードでは、ユーザが入力した文字をそのまま MS-DOS に渡していく。変換を行いたいときは、まず変換処理に入ることを示すキーを入力してから文字列を入力する。システムは、文字列

表 3 変換システムの性能評価 (2)

Table 3 The performance of conversion systems (2).

RAM ディスク使用		辞書引き 1 回	辞書引き 2 回	辞書引き 3 回	辞書引き 4 回
従来の仮名漢字変換		134 ミリ秒 (100)	202 ミリ秒 (100)	237 ミリ秒 (100)	294 ミリ秒 (100)
交ぜ書き辞書法	仮名入力	171 ミリ秒 (128)	262 ミリ秒 (130)	297 ミリ秒 (125)	352 ミリ秒 (120)
	漢字混じり入力	250 ミリ秒 (187)	326 ミリ秒 (161)	391 ミリ秒 (165)	418 ミリ秒 (142)
漢和辞書法	仮名入力	138 ミリ秒 (103)	203 ミリ秒 (100)	238 ミリ秒 (100)	297 ミリ秒 (101)
	漢字混じり入力	159 ミリ秒 (119)	275 ミリ秒 (136)	364 ミリ秒 (154)	433 ミリ秒 (147)

表 4 変換システムの性能評価 (3)

Table 4 The performance of conversion systems (3).

ハードディスク使用		辞書引き 1 回	辞書引き 2 回	辞書引き 3 回	辞書引き 4 回
従来の仮名漢字変換		178 ミリ秒 (100)	243 ミリ秒 (100)	288 ミリ秒 (100)	346 ミリ秒 (100)
交ぜ書き辞書法	仮名入力	325 ミリ秒 (183)	387 ミリ秒 (159)	409 ミリ秒 (142)	510 ミリ秒 (147)
	漢字混じり入力	421 ミリ秒 (237)	477 ミリ秒 (196)	540 ミリ秒 (188)	463 ミリ秒 (134)
漢和辞書法	仮名入力	174 ミリ秒 (98)	244 ミリ秒 (100)	289 ミリ秒 (100)	349 ミリ秒 (101)
	漢字混じり入力	243 ミリ秒 (137)	364 ミリ秒 (150)	442 ミリ秒 (153)	497 ミリ秒 (144)

をバッファにため、変換キーが押された時点で変換を行う。ユーザが変換結果を正しいものとして確定すると、それ以降は、ユーザが入力した文字をそのまま MS-DOS に渡す状態に戻る。

この二つの入力モードは、ユーザの漢字コードの習熟度によって使い分けられることを想定したもので、漢字コードを十分に覚えているユーザは、直接入力モードで次々に入力していける。彼らが、間接入力モードを用いると、いわゆる無変換キーばかり押ししていなければならない。彼らにとって、仮名漢字変換は、漢字コードを覚えていないか忘れてしまった漢字を入力するときだけ用いるものである。ちょうど、手書き文書作製時に、知らない漢字だけを国語辞典で調べて書くようなものである。一方、TUT コードの初心者には、間接入力モードで、常時（漢字混じり）仮名漢字変換を用いながら、入力していけばよい。

TUT コードのための仮名漢字変換システム固有の二つ目の機能は、変換結果の漢字コードの表示である。例えば、「にほん」と入力して変換し、「日本」を得たときに、CRT 画面最下行右端に、その漢字コードである「kd ud」を表示する。1章にも述べたように、漢字混じり仮名漢字変換は、漢字コードの覚えはじめのユーザの便宜を図ることが、目的の一つである。本機能は、こうしたユーザのためのものである。

なお、作製したデバイス・ドライバは、以下のアプリケーション・システムの日本語フロントエンド・プロセッサとして動作することを確認した。

- (1) エディタ: RED++, MIFES 98
- (2) 日本語ワープロ: 新松
- (3) データベース: dBASE III
- (4) 表計算ソフト: Lotus 1-2-3

5. おわりに

2 ストローク日本語入力方式に適した仮名漢字変換として漢字混じり仮名漢字変換を提案し、二つの漢字混じり仮名漢字変換方式—交ぜ書き辞書方式と漢和辞書方式—を、MS-DOS 上にデバイス・ドライバとして実現した。実現した二つのシステムを評価した結果、デバイス・ドライバの大きさや変換速度に関しては、両者とも実用に供しうるものとなっている。ただし、辞書の大きさの観点から評価すると、現在のパソコン環境では、漢和辞書方式のほうが、より実用性が高いことが明らかになった。

謝辞 本研究に関し討論いただいた豊橋技術科学大

学情報工学系大岩研究室の諸氏に感謝します。

多くの有益なご助言をいただいた査読者の方々に感謝します。

参 考 文 献

- 1) 山田尚勇: タイプライタの歴史と日本語入力, 情報処理, Vol. 23, No. 6, pp. 559-564 (1982).
- 2) 大岩 元, 高嶋孝明, 三井 修: 日本語タッチタイプ入力の一方式, 情報処理学会論文誌, Vol. 24, No. 6, pp. 772-779 (1983).
- 3) 三橋一夫: 新日本語入力術, pp. 96-100, JICC 出版局, 東京 (1988).
- 4) 山田尚勇: コード入力方式と人間性, ASCII Vol. 10, No. 12, pp. 236-237 (1986).
- 5) 小野芳彦: T コードの補助入力: 字形組み合わせ法と交ぜ書き変換法, 情報処理学会論文誌, Vol. 31, No. 3, pp. 404-414 (1990).
- 6) 喜多辰臣, 宮崎充美, 河合和久, 大岩 元: 漢字混じり仮名漢字変換, 昭和 61 年度電気関係学会東海支部連合大会, p. 507 (1986).
- 7) 喜多辰臣, 塩見彰睦, 河合和久, 大岩 元: 2 ストローク入力仮名漢字変換システム, 情報処理学会文書処理とヒューマンインタフェース研究会, 16-4 (1988).
- 8) 国立国語研究所: 電子計算機による新聞の語彙調査, 秀英出版, 東京 (1970).
- 9) 中津山秀範: VJE- α について, Oh! 16, Vol. 2, No. 9, pp. 43-48 (1985).
- 10) 日経バイト (編): 拡張進む MS-DOS, 日経バイト, No. 52, pp. 152-177 (1988).

(平成 3 年 9 月 2 日受付)

(平成 4 年 5 月 14 日採録)

塩見 彰睦 (正会員)

1965 年生。1985 年徳山工業高等専門学校情報電子工学科卒業。同年豊橋技術科学大学情報工学課程 3 年次編入。1989 年同大学院工学研究科情報工学専攻修了。工学修士。

現在、同大学院工学研究科博士後期課程に在学中。ソフトウェア工学, HCI, 日本語入力の研究に従事。CASE 環境, HCI に興味をもつ。日本ソフトウェア科学会会員。

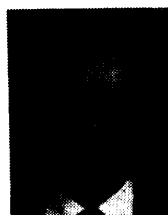
喜多 辰臣 (正会員)

1964 年生。1986 年豊橋技術科学大学情報工学科卒業。1988 年同大学院修士課程修了。工学修士。同年富士ゼロックス(株)入社。日本語入力・ユーザインタフェース等の設

計・開発に従事。

**河合 和久 (正会員)**

1958年生。1981年大阪大学基礎工学部情報工学科卒業。1986年同大大学院基礎工学研究科博士課程修了。工学博士。同年豊橋技術科学大学情報工学系助手。1991年同大知識工学系講師。現在に至る。計算機を用いた教育、創造的活動における計算機支援、HCIなどの研究に従事。電子情報通信学会、人工知能学会、AAAI、日本ソフトウェア科学会などの会員。

**大岩 元 (正会員)**

1942年生。1965年東京大学理学部物理学科卒業。1971年同大学院博士課程修了。理学博士。同年東京大学理学部助手。1978年豊橋技術科学大学情報工学系講師。1980年同助教授。1985年同教授。1992年慶應義塾大学環境情報学部教授。1974年～1976年英国ケンブリッジ大学キャベンディッシュ研究所客員研究員(ブリティッシュ・カウンシル・スカラ)。1979年～1980年米国コーネル大学応用物理学科客員準教授。キーボード入力、情報教育、ソフトウェア工学などの研究に従事。