

平成14年度

筑波大学第三学群情報学類

卒業研究論文

題目 ペンによるメニュー選択に基づく  
日本語入力手法

主専攻 情報科学

---

著者名 佐藤 大介

---

指導教員 電子・情報工学系 田中 二郎

---

# 要 旨

我々はペン型デバイスによる新しい日本語入力システムを提案し、実装した。本システムにおいて、ユーザは子音による入力を行い、その入力に対して予測される候補を選択することで日本語を入力することができる。また、目的の候補がない場合など必要に応じて母音の選択をすることで、絞込みを簡単に簡単に行える。

子音の入力と候補の選択はドーナツ型に表示されたメニューのメニュー選択を行うことで実現される。このインタフェースにより、ユーザは候補の選択と子音の入力を連続的な操作で行えるようになった。また、メニューにおける子音の位置を覚えることで、子音の入力をペン型デバイスの軌跡を目で追わずに素早く行えるようになった。

# 目次

第1章	はじめに	1
第2章	フローメニューとペンによる文字入力	3
2.1	フローメニュー	3
2.1.1	フローメニューの操作例1	4
2.1.2	フローメニューの操作例2	4
2.2	ポップアップメニューとフローメニューの比較	5
2.2.1	ポップアップメニュー	5
2.2.2	比較	6
2.3	ペンによるアルファベット入力	7
2.3.1	Quikwriting	9
2.4	さまざまな日本語入力手法	11
2.4.1	予測を用いた日本語入力	13
2.4.2	子音入力による日本語入力	13
2.4.3	ペンによる日本語入力	13
第3章	日本語入力システム Popie	15
3.1	本章の概要	15
3.2	Popie の特徴	15
3.3	子音入力における変換候補	15
3.4	Popie のインタフェース	16
3.4.1	子音の入力	16
3.4.2	母音の指定	17
3.4.3	候補の表示と選択	18
3.4.4	キャレット位置の調節	20
3.4.5	アンドゥー・リドゥー	21
3.4.6	記号や英数字の入力	22
3.4.7	Popie のメニュー配置	22
3.5	Popie の実装	23
3.5.1	辞書と検索	23
3.5.2	左利き	25
第4章	システムの実装	26
4.1	実装環境と言語	26
4.2	クラス階層	26

4.2.1	オブジェクトのクラス群 . . . . .	26
4.2.2	フローメニューのクラス群 . . . . .	27
4.2.3	日本語処理のクラス群 . . . . .	27
4.2.4	アンドウ・リドウのクラス群 . . . . .	27
<b>第 5 章</b>	<b>システムの評価実験</b>	<b>29</b>
5.1	実験方法 . . . . .	29
5.1.1	被験者 . . . . .	29
5.1.2	方法 . . . . .	29
5.1.3	実験環境 . . . . .	29
5.1.4	入力する文章 . . . . .	29
5.2	実験結果 . . . . .	30
5.3	評価 . . . . .	30
<b>第 6 章</b>	<b>まとめ</b>	<b>34</b>
	<b>謝 辞</b>	<b>35</b>
	<b>参 考 文 献</b>	<b>36</b>

# 第1章 はじめに

近年プラズマディスプレイのような大型の表示機器が登場し、会議やミーティングといった場面において電子ホワイトボードとして利用する事ができるようになった。また、情報提供や広告を目的とするパブリックディスプレイとしてこれから多く普及することも予想される。

このような大型の表示機器を電子ホワイトボードとして使用した際のコンピュータ操作は、これら画面の近くで行うことが多い。このとき、マウスとキーボードが画面の近くにあったとしても画面を見上げる格好になったり、立ったままマウスとキーボードを使うことになり、操作しにくいことが問題となる。こういった状況において、ペン型デバイスや指で直接画面に触れる方が操作を行い易く、また画面に直接触れることで操作が直感的になるのでマウスとキーボードを使うよりも、ペン型デバイスを使った方が良いと考えられる。今後、PDA やタブレット PC のようなペン型デバイスで操作するコンピュータは、ますます増えると予想されており、ペン型デバイスによる入力はさらに重要になるであろう。

しかし、従来のペン型デバイスによるコンピュータ操作は、マウスとキーボードのデバイスの操作を1つのデバイスで単にエミュレートするだけである。

ペン型デバイスによるマウス操作のエミュレートは、タップ（操作面を1回だけ突っつくこと）がクリックに相当し、ダブルタップがダブルクリックに相当する。また、ペン型デバイスに取り付けられたボタンを押しながらタップする事が右クリックに相当している。だが、ペン型デバイスはマウスに比べて一箇所に固定するのが難しくペン先が移動し易いので、タップやダブルタップ操作はマウスでクリックやダブルクリックをするよりも難しい操作になってしまう。

ペン型デバイスによるキーボード操作、つまり文字入力は、ソフトウェアキーボードの様に、操作面に表示されたボタンを押すことによりキーボードをエミュレートするものであったり、Graffiti[Bli95] に代表されるような文字認識をキーボードの打鍵の代わりに用いるものがほとんどである。しかしながら、ソフトウェアキーボードを用いたインタフェースの場合、細かいボタンをペン型デバイスで押す必要があるので、ボタンの位置を目で追わなければならない。入力スピードも上がらない。文字認識を用いた日本語の入力でも、Graffiti であればローマ字入力になるのでストローク数が多くなり、且つ変換の手間もかかってしまう。かな漢字の文字認識の研究も数多くなされているが、崩し字や続け字のような文字に対しては、認識性能が十分であるとは言えない段階である [森 02]。

さらに、このようなペン型デバイスによる既存の入力環境において、我々が最も問題視する点はペン型デバイスによるメニュー操作と文字入力操作がまったく別のインタフェースで構成されるということである。一般的な GUI の操作は、1) クリックやドラッグ等の操作、2)

メニュー操作, 3) 文字入力に分けられる。マウスとキーボードを使う環境では 1), 2) を主にマウスで, 3) を主にキーボードで行う。キーボードとマウスを使用する環境において, ユーザはキーボードとマウスの操作を切り替えるために手を動かす必要があり, 交互に作業する場合は非常に面倒であるという問題があった。

ペン型デバイスを使用する場合においても, 1), 2) を行うとき, 3) を行うときでメニューやソフトウェアキーボードとの間で行ったり来たりを繰り返すことになり, ペンの移動距離が増えてしまう。特に大型の画面で操作を行うときにはインタフェースが物理的により離れることになり, より顕著な問題となる。ペン型デバイスを使用する場面において, この問題を解決するためには, 1), 2), 3) の操作を統一したインタフェースの開発が必要である。これらの操作の中で 1) の操作は画面上のどこの位置でも行えることが求められる操作であるので, 2), 3) の操作は画面上どこでも行える必要がある。

F. Guimbretiere らによる FlowMenu [GW00] は連続的で滑らかな入力が可能なメニューで, ペン型デバイスで画面に触れた位置を中心にして表示されるドーナツ型のメニュー(以降フローメニューと表記する)であり, 画面のどの位置でも表示してメニューの選択を行うことが可能である。また, フローメニューでは Quikwriting [Per98] によってアルファベットや記号の入力が行える。

フローメニューを用いることで, 操作を統一し上記の問題を解決できると考えるが, Quikwriting には日本語のような変換を伴う言語の入力はできないという問題点がある。そこで, 我々はフローメニュー上で日本語入力可能なシステムを提案し, 実装した。我々のシステムでは, ユーザは子音を入力し, その入力に対して予測される候補を選択することで日本語を入力する。また, 目的の候補が出現しない場合や, 辞書に未登録の単語を入力するときなど, 必要に応じて母音の選択も簡単に行えるようになっている。子音の入力を用いることで, ユーザが入力する文字を減らすことができるため, ユーザへの負担を減らすことができる。また, 新聞の記事データ [毎日 01] を元に作成した辞書や, ユーザの入力から学習した辞書を用いて入力に対する候補を予測して少ない入力文字数の補助をしている。

本論文の構成は以下のようになっている。まず第 2 章で, フローメニューと, さまざまな文字入力システムを紹介し, ペンによる日本語入力について述べる。次に, 第 3 章では, 我々の実装したフローメニュー上での日本語入力システム Popie の詳細と実装について述べ, 第 4 章では Popie を含むシステム全体の実装について述べる。第 5 章ではユーザによる評価実験によってシステムの評価し, 第 6 章でまとめる。

## 第2章 フローメニューとペンによる文字入力

### 2.1 フローメニュー

フローメニュー（図 2.1）は、ペン型デバイス（以降ペンと表記する）のみを入力デバイスとして用いるシステムのためのメニューである。F. Guimbretiere らのシステム [GW00] では、ペンについてのボタン、もしくはペンとは別のデバイス上のボタンを押したまま、操作面にペンで触れることにより、メニューを表示させている。

フローメニューは、ペンによって画面に触れた位置の周りにドーナツ状に表示されるようになっていいる。表示されるドーナツ状のメニューは上下、左右と斜め方向に 45 度ずつ 8 等分に分割されている。8 分割されたドーナツ状の領域のそれぞれの事をオクタント (octant) と呼び、8 つのオクタントの内側にある中心部分の領域をレストエリア (rest area) と呼んでいる。

フローメニューでは表示された初期状態において、ペンの先はレストエリアの中心にある。ユーザはその位置から、メニュー項目に対応したオクタントにペンを移動し、再びレストエリアにペンを戻すことでメニューの選択を行う。このとき、ユーザはレストエリアにペンを戻す前に、操作面からペンを離すことによりメニュー選択の操作をキャンセルすることができる。

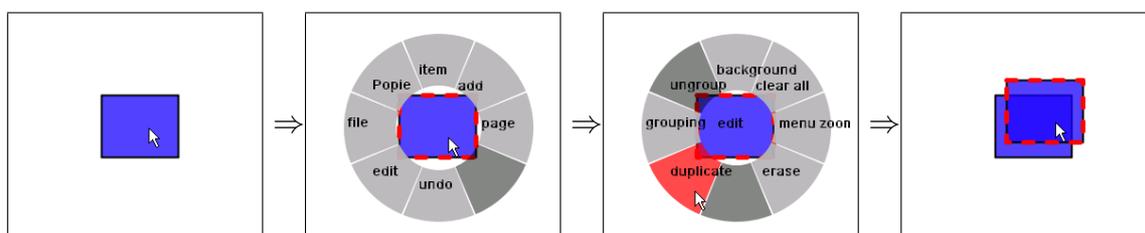


図 2.1: フローメニュー

フローメニューの初期状態では 8 つのトップレベルのメニューがあり、それぞれのオクタントに対応している。それぞれのオクタントにペンを移動することにより、さらに 8 つずつのサブメニューが表示される。オクタントからレストエリアにペンを戻すとき、ペンが最後にあったオクタントのサブメニューが選択される。フローメニューでは、オクタントとレストエリアをペンが行ったり来たりすることで、さらに深い階層のメニューを実現することが可能である。

以下にフローメニューの操作例を 2 つ挙げる。

### 2.1.1 フローメニューの操作例 1

1 つめの操作例として、オブジェクトを削除する例を図 2.2 に示す。この例では青、赤、黄のオブジェクトのうち、赤色のオブジェクトを削除する。まずは操作対象である赤いオブジェクトの上でペンをタップ&ホールドし、フローメニューを表示する。表示されたメニューの左下方向にある“edit”とラベル付けされたオクタントにペンを移動させることにより“edit”メニューを選択する。“edit”メニューを選択すると表示が切り替わり、“grouping”や“erase”といったメニュー項目が現れる。ここで“erase”メニューのオクタントまでペンを移動させた後、レストエリアにペンを戻すことにより“erase”メニューを選択し、オブジェクトを削除することができる。

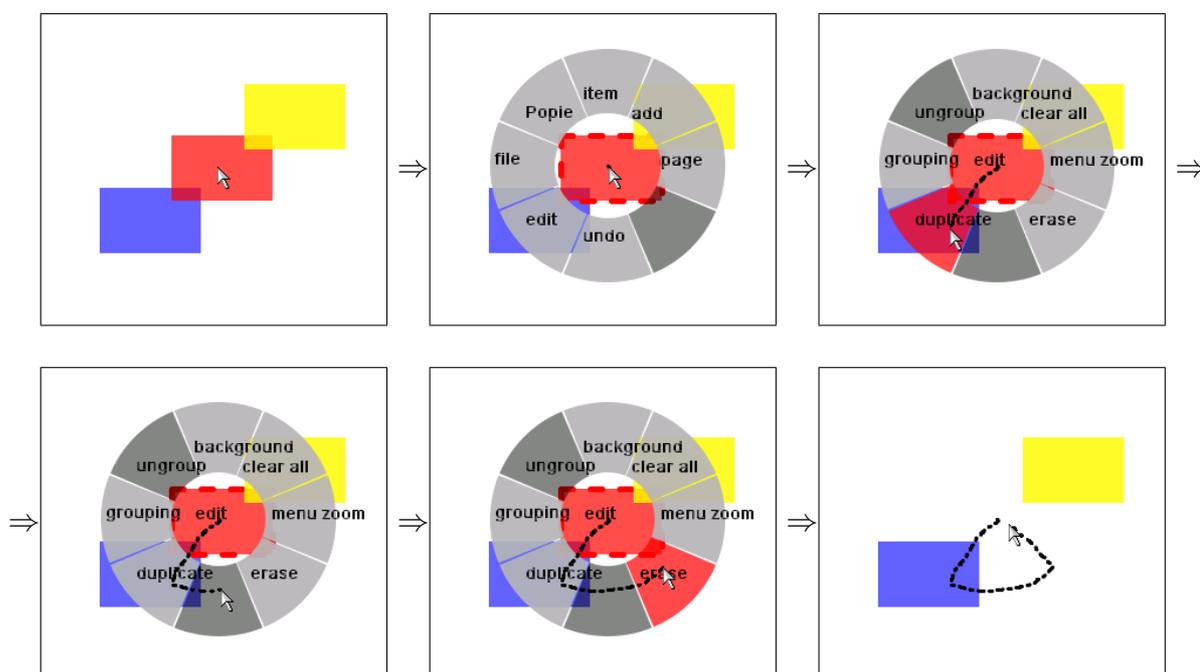


図 2.2: フローメニューによる erase 操作の例

### 2.1.2 フローメニューの操作例 2

フローメニューでは操作例 1 で挙げたようなオクタントとレストエリアを行ったり来たりする基本的なメニュー操作の他に、フローメニューをダイアルのようにして使用方法がある。ダイアル操作としてオブジェクトの回転の例を図 2.3 に示す。まず最初に対象となるオブジェクトの上でフローメニューを表示したのち、“item”メニューから“rotate”メニューを選択し、その後表示されるダイアル状のメニューで回転角度を調整してペンを離して決定する。

このようにして、フローメニューをダイアルとして使用することにより回転角度やズーム率、あるいは縦、横方向の位置などを一定の操作量で細かく操作することが可能になる。ダイアルとしてフローメニューを使うときは、オクタントとからオクタントへの移動が、動作のトリガとなっている。

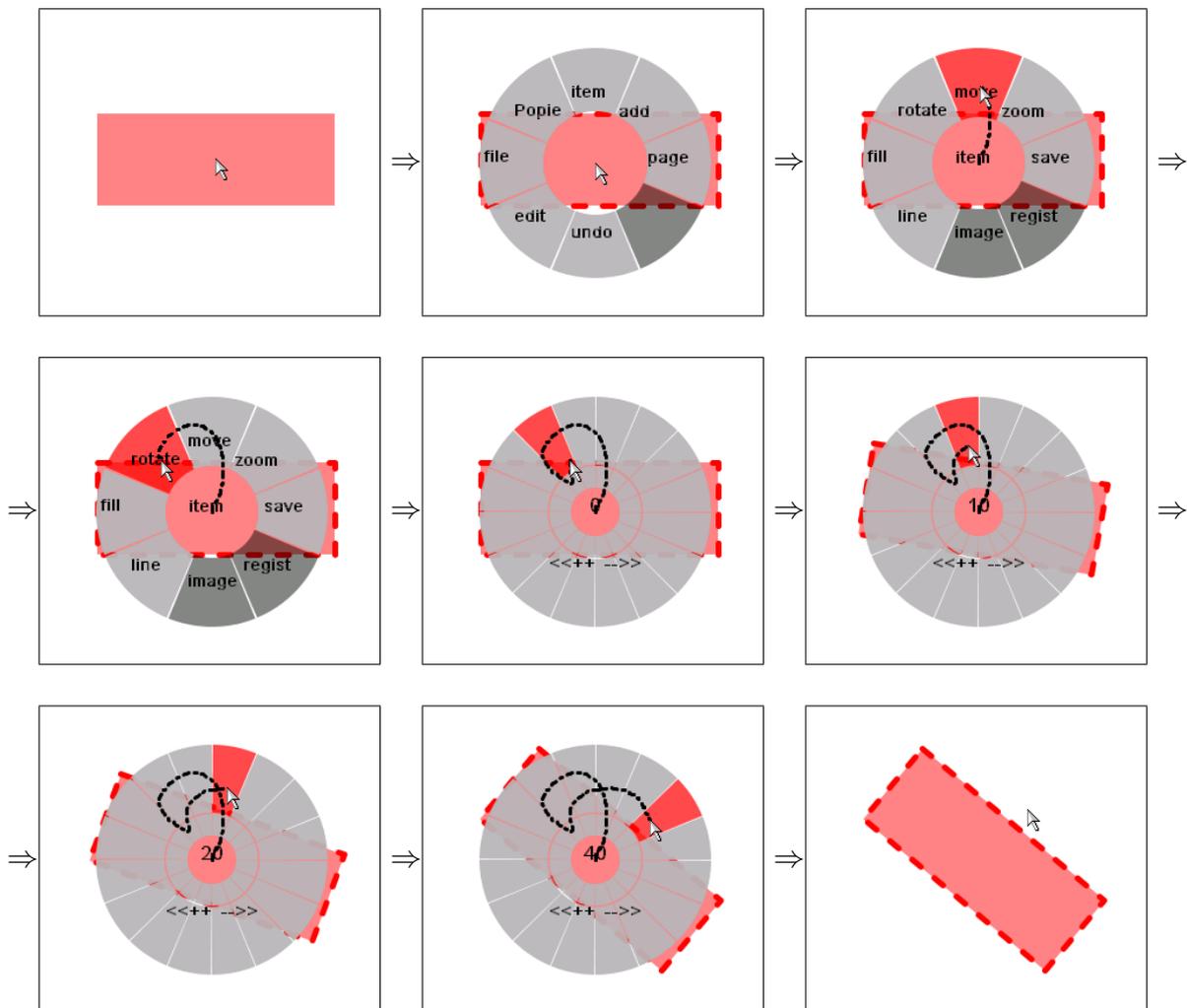


図 2.3: rotate 操作

## 2.2 ポップアップメニューとフローメニューの比較

次に、ポップアップメニューとフローメニューの比較について述べるが、先にポップアップメニューについて整理しておく。

### 2.2.1 ポップアップメニュー

ポップアップメニュー（図 2.4）は、今日のコンピュータの GUI 環境において広く用いられているメニューである。ポップアップメニューはメニューバーのメニュー項目をクリックしたり、右クリックなどのトリガを用いて表示させることができる。ユーザは、表示された矩形の領域に並べられたメニュー項目の中の目的の項目をクリックすることでメニューを選択する。

また、メニュー項目の選択をトリガとして新たなメニューを表示することにより、1 階層深いメニュー選択を行うことができる（図 2.5）。この繰り返しによって深い階層のメニュー

選択が可能である。

ポップアップメニューの問題点としては、メニューの表示位置がユーザが操作する場所によって変化するということである。例えば、普段はクリックした右側に表示されるメニューが画面の右端の方では左側に表示されるような場合である。このようにポップアップメニューは常に同じ位置に表示されるとは限らないので、ユーザがメニュー項目の位置を勘違いして操作を間違えてしまう原因となる。

ポップアップメニューでは、一度に多くのメニュー項目を表示ができるという点で優れている。しかし、メニュー項目が多すぎるとメニューが画面を広く占有してしまうばかりでなく、ユーザが目的のメニューを見つけるのに時間がかかってしまうことも問題である。



図 2.4: ポップアップメニュー



図 2.5: 階層が深くなったときのポップアップメニュー

## 2.2.2 比較

ポップアップメニューとフローメニューの違いを表 2.1 に示す。

フローメニューでは階層が深くなっても最初に表示した領域を占有するだけで、ポップアップメニューのように画面を大きく占有することがない。このことにより、メニュー選択時のペンの移動を少なくすることができる。また、フローメニューではポップアップメニューのように、ユーザが操作する位置によってメニューの表示位置が変化しないので、メニュー項目の位置を記憶することでメニューの入力が高速に行えるようになる。例えば“Move”メニュー

	ポップアップメニュー	フローメニュー
メニュー選択方法	マウスのクリック or リリース	オクタントとレストエリアの境界線を越えることによる
メニュー項目の数	画面のサイズが許す限り表示可能	最大で 8 個のメニューを表示可能
メニューの表示領域	階層が深くなるとどんどん広がってしまう	階層が深くなっても表示される領域には変化がない
メニュー項目の位置	ユーザが操作する場所によって表示位置が変わってしまうので、メニューとポインタの位置を常に確認しながら操作する必要がある。	上下、左右などの位置は変わることがないのでメニューを位置で覚えることができる。場所を覚えることでメニューを見なくても操作ができるようになる。

表 2.1: ポップアップメニューとフローメニューの違い

を選択するという操作（図 2.6）では、“Item”メニューの中の“Move”メニューを選択する操作になる。このとき“Item”メニューは初期状態の上方向のオクタントにあり、“Move”メニューも“Item”メニューの上方向のオクタントにある。つまり“Move”メニューを選択するには、メニューが表示されてから上方向のオクタントにペンを移動したのちに、再びレストエリアに戻せば良い。

これは一種のジェスチャであると考えられる。この場合フローメニューはそのジェスチャーの誘導の表示として働く。例えば、ユーザは“Move”メニューの位置を覚えることにより、メニューを見なくても、“Move”メニューを選択することができる（図 2.7）。

一方ポップアップメニューでは、メニューの表示位置が常に同じではなかったり、一度にたくさんのメニュー項目が表示されてしまうため、目的のメニュー項目の位置とポインタの位置を常に目で追いかける必要があり、フローメニューのように素早いメニュー入力を行うことはできない。

## 2.3 ペンによるアルファベット入力

ペンを用いたアルファベット入力手法には様々なものがある。表 2.2 にペンを用いたアルファベット入力手法とその特徴を示す。

ペンを用いたアルファベット入力は大きく 3 つの形式に分類することができる。1 つ目はキーボード型で、画面に擬似的なキーボードを表示し、そのキーをペンで押すことにより文字を入力する方法である。この方法においての入力速度は、キー配列が最も重要な要因となる。一般的によく使われる Qwerty 配列のキーボードの他に、ペンで入力した際にペンの移動が最小限に抑えられるようにキーが配列された Metropolis [ZHS00] や OPTI [MZ99], FITALY [Sol98] などがあり、Qwerty 配列を用いた場合よりも速い入力速度を得られている。また、MessageEase [EXi] のようにテンキーの配列を用いて入力する方法もある。この方法

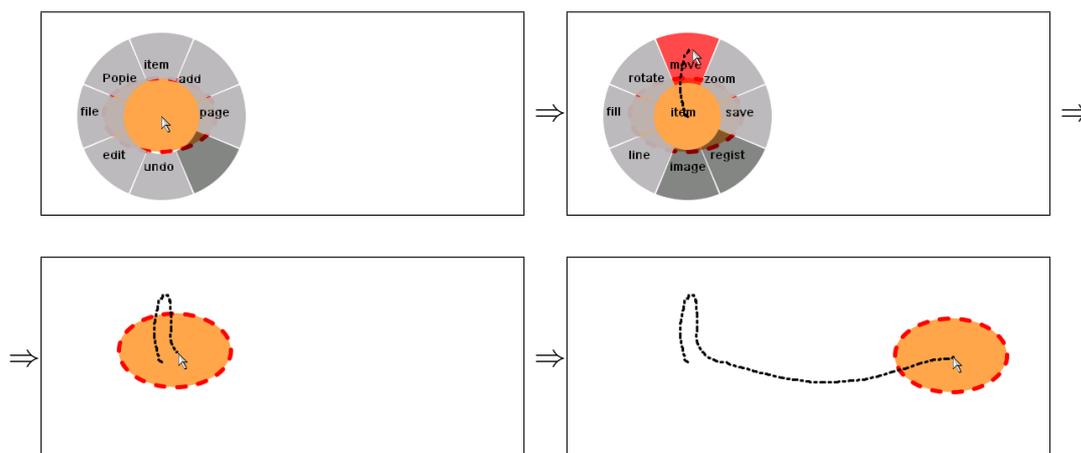


図 2.6: 普通の move 操作

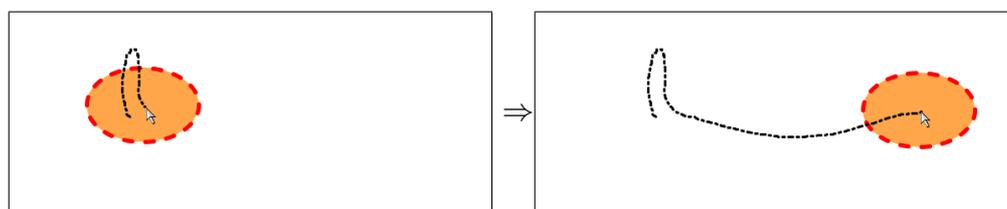


図 2.7: メニューが表示されない場合の move 操作のイメージ

では、1つまたは2つのキーを組み合わせることで文字を入力することになる。

2つ目は、文字を書くようにして入力するストローク型である。Graffiti [Bli95] や Unistroke [GR93] は1ストロークでアルファベットを入力する方法である。Graffiti ではアルファベットとほぼ同じストロークを使うが、Unistroke では若干アルファベットと異なるストロークを使う。このため Unistroke は Graffiti より覚え難くなるが、Unistroke でのストロークは Graffiti のそれよりも短くてすむので、覚えてしまえば Graffiti よりも Unistroke の方が早く入力が可能になる。図 2.8 に Graffiti と Unistroke のストロークを示す。

3つ目は、メニュー型の入力方法である。これはフローメニューにおけるメニュー選択のようにある領域からある領域に行ったり来たりすることで文字を入力する方法である。Quikwriting [Per98] はフローメニューと同じく中心部分と8分割された領域によって文字入力を行う方法である。この方法ではストロークを連続させて入力を行う事ができる。また、フローメニューでメニュー項目を位置で覚えたように、文字を位置で覚えることで素早い入力が可能になる。T-Cube [VN94] はパイメニューを用いた入力方法で、初期状態ではフローメニューとほぼ同じであるが、オクタントにペンを移動させた時に新しい円形のメニューをペンの位置に表示する点が違う。この方法では文字の入力はペンを離すことにより行うので、1文字入力するごとにペンを操作面から離す必要がある。Cirrin [MA98] はフローメニューに似ているが円を8等分ではなく、26等分したインターフェースを使用する入力方法である(図 2.9)。それぞれの領域がアルファベットに対応しているが、1つ1つの領域が小さくなってしまっているので、文字を位置を覚えても結局ペンを目で追わないと入力が難しくなっている。

入力の形式	システム名	特徴
キーボード型	Qwerty	一般的なキーボードの配列. 最も良く使われているが, 最も最適な配列でないことは知られている.
	Metropolis [ZHS00] OPTI [MZ99] FI- TALY [Sol98]	ペンを用いて入力をするために最適化した配列. Metropolis はキーが蜂の巣のように六角形になって配置されており, OPTI, FITALY よりも高速な入力が可能である.
	MessageEase [EXi]	テンキー上に配置された文字を入力する. 頻出アルファベットはそれぞれの数字に, 残りのアルファベットは二つのテンキーを組み合わせることにより入力する.
ストローク型	Graffiti [Bli95]	PDA のような小型デバイス上で良く使われているシステム. アルファベットを1ストロークで描くことにより文字を入力する. ストロークは元のアルファベットに近い.
	Unistroke [GR93]	Graffiti に似ているが Graffiti よりもストロークが短くなる. ストロークは元のアルファベットとは若干異なる.
メニュー型	Quikwriting [Per98]	中心とその周りに8方向に分割された領域上でペンを移動させることにより文字入力が可能. ペンを操作面から離さなくていい.
	Cirrin [MA98]	Quikwriting に似ているが, 26個に分割されていて細かいので文字の選択が難しい.
	T-Cube [VN94]	パイメニューを使った文字入力. 文字を選択するときにペンを操作面から離す必要がある.

表 2.2: ペンを用いた文字入力手法

次節では, フローメニューと同じ方法で動く Quikwriting について細かく説明する.

### 2.3.1 Quikwriting

前節で述べたように Quikwriting は, フローメニューのように8方向に分割された領域を使ってアルファベットや記号を入力するシステムである. このシステムはPDAなどのペンを用いたデバイス上で使うために開発されたものである. Quikwriting は Graffiti に比べて入力するときのストロークが短く, ペンを入力対象のデバイス上から離す必要がないので, Graffiti よりもストレスを感じることなくスムーズな入力ができるようになっている.

#### Quikwriting のインタフェース

本論文で紹介する Quikwriting には現時点で入手可能な一番新しいバージョンのデモシステムを用いた.

Quikwriting は図 2.10 のようなインタフェースを持つシステムである. 左半分には主にア

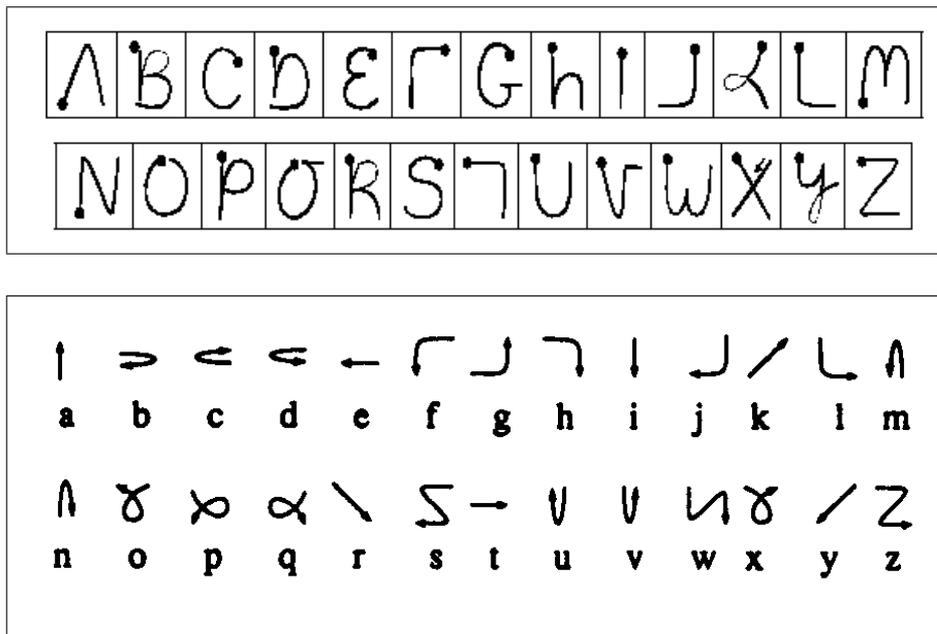


図 2.8: Graffiti のストローク (上) と Unistroke のストローク (下)

ルファベット, 右半分には主に数字や記号が含まれる. 基本的には一番外側の枠に表示されている文字を入力し, 内側の記号は特別な記号を用いて入力する.

‘ ’, ‘ ’, および ‘ ’ は特別な記号で, その記号を入力した次の入力のみ反映される修飾記号である. それぞれの記号の意味は以下の通りである.

- ‘ ’ は内側に表示された記号を選択するための記号
- ‘ ’ は大文字を入力するための記号
- ‘ ’ は左右どちらで操作をしても右半分の入力を行ったものとする記号

また, ‘ ’ は改行, ‘ ’ はバックスペース, ‘ ’ はスペースを入力する.

### Quikwriting による文字入力例

Quikwriting の文字入力例を以下に示す.

‘f’ を入力する場合は図 2.11 のように, 右上のオクタントに “xfn” と並んでいるので右上のオクタントにペンを移動させたのち, 上方向のオクタントにペンを移動させてからレストエリアにペンを戻すという操作をする.

同様に ‘x’ を入力する場合は右上のオクタントにペンを移動させたのち, 左上のオクタントにペンを移動させてからレストエリアにペンを戻す. また ‘n’ を入力する場合は右上のオクタントにペンを移動させたのち, そのままレストエリアにペンを戻す. このように, 表示位置と操作とを対応付けてある.

“The” を入力するときは ‘ ’, ‘t’, ‘h’, ‘e’ の順に入力を行う.

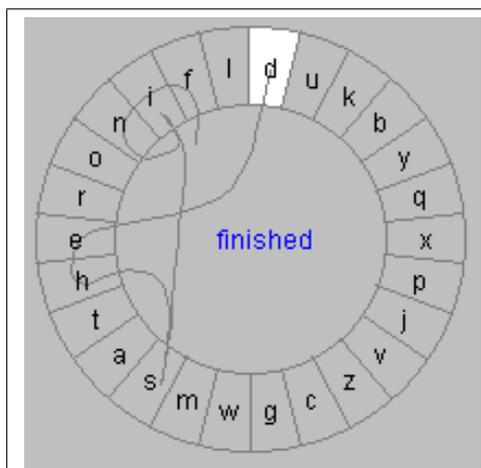


図 2.9: Cirrin のインタフェース

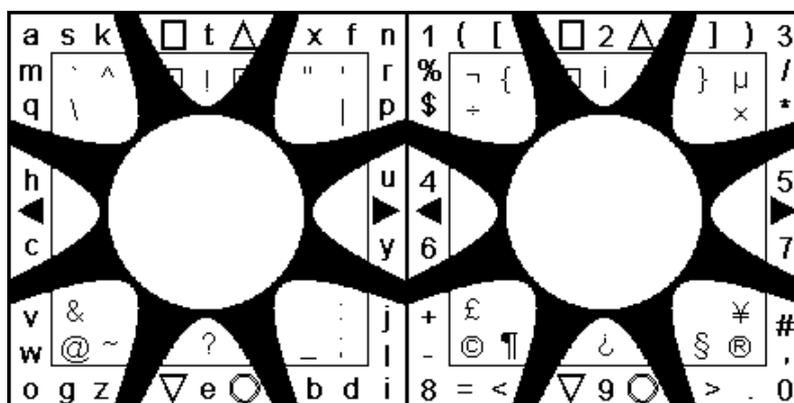


図 2.10: Quikwriting のインターフェース

‘ ’は上方向のオクタントに“ t ”とあるので、上方向のオクタントにペンを移動させたのち、右上のオクタントにペン移動させてレストエリアにペンを戻す。‘t’は上方向のオクタントに“ t ”とあるので、上方向のオクタントにペンを移動させたのち、レストエリアにペンを戻す。‘h’は左方向のオクタントに縦に“h c”とあるので、左方向のオクタントにペンを移動させたのち、左上のオクタントにペンを移動させてレストエリアにペンを戻す。‘e’は下方向のオクタントに“ e ”とあるので、下方向のオクタントにペンを移動させたのち、レストエリアに戻す。ペンの軌跡は図 2.12 (右) のように連続的なものとなる。

ここで説明した“The”の入力例のように、Quikwriting におけるそれぞれの英単語の入力のは、一つのジェスチャーのように入力することが可能になる。

## 2.4 さまざまな日本語入力手法

ペンによる日本語入力を考える前に、日本語入力について整理する。

日本語入力に関する研究は数多くなされおり、さまざまな手法が考えられている。それらのいくつかとその特徴を表 2.3 にまとめた。

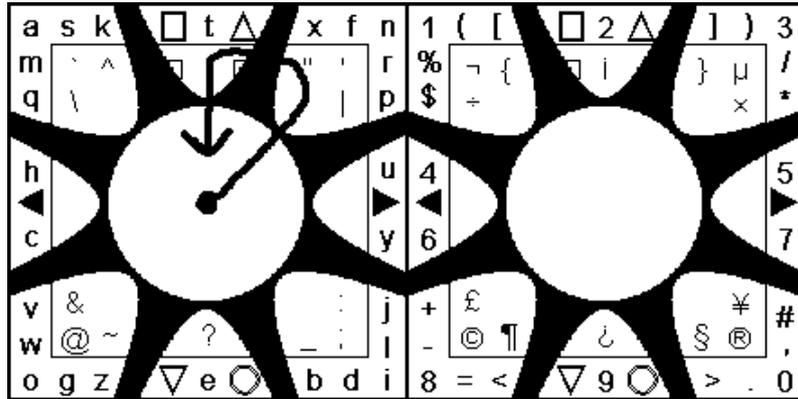


図 2.11: ‘f’ を入力

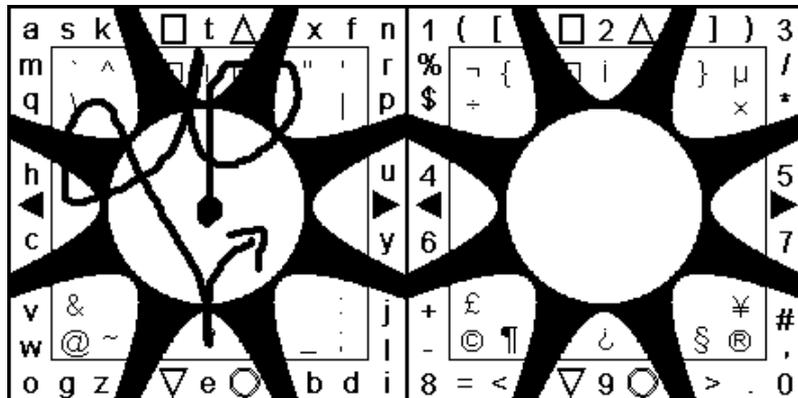


図 2.12: “The” を入力

日本語入力を行うプロセスは大きくわけて2つある。1つは文章の読みを入力することと、もう1つは、その読みを変換することである。読みを入力する方法には、ローマ字入力や1ひらがなを1タイプで入力するJISかな入力がある。この他には2タイプでかなを入力する方法 [富樫 99a] [増田] [森田 98] などがある。読みの変換を行うプロセスは、最後まで読みを入力してから変換する方法と、読みの入力途中で予測された候補を選択する方法 [増井 97] [小松 02] [TOS] がある。

また特殊な入力方法としては、かなと漢字を両方ともキーボードから直接入力する方法 [森田 97] [増村 97] [富樫 99b] や、子音のみの入力から予測された候補を選択する方法 [T9] [TIYM01] がある。

最近多くなってきた入力方法としては、ユーザが書いたかな漢字の文字を認識する方法である。この方法は変換の必要もなく、書いた文字がそのまま入力されるのもっとも自然な入力方法である。しかしながら、現在の段階では決められた枠の中に文字を楷書で書く必要があり、ストローク数も増えてしまい入力速度は速くない。

このように日本語入力手法には様々なものがあるが、本研究の対象であるペンを用いた入力の場合、操作面を直接操作できることで、より直感的な操作が可能であると引き換えにユーザの操作する手の移動量が大きくなり、キーボードによる入力よりも入力速度が劣って

しまう。そこで、より少ない入力で目的の文章を打てるような工夫が必要になる。以下には、そのような目的で研究開発されている日本語入力システムについて説明する。

#### 2.4.1 予測を用いた日本語入力

POBox [増井 97] や MobileRupo [TOS] などの予測を用いたシステムは、主に携帯電話や PDA などの小型のデバイスにおける入力手法として採用されている。

これらのシステムはユーザが入力する文字列から予測される単語を候補として挙げたり、入力された単語の次に入力されると予測される単語を候補として挙げることにより、実際に入力する文字の数を少なくし入力の効率を上げようとするものである。例えば、ユーザがすでに“よろしく”と入力を済ませていた場合を考える。次にユーザが‘お’と入力したら“お願いします”、“お願い致します”とユーザが入力するものと予測して候補にこれらの単語を表示するという仕組みである。これらのシステムではユーザが使い込むほどにユーザに適応した辞書ができるので、個人で使うことの多い小型デバイス上での入力システムとしてはとても良い。

#### 2.4.2 子音入力による日本語入力

Touch me key 10 [TIYM01] や T9 [T9] などは子音による入力から候補を予測し日本語を入力するシステムである。T9 は携帯電話の入力方法として採用されている。

子音による入力は、例えば“田中”と入力するためにローマ字では“TANAKA”と入力するが、‘A’を取り除いた“TNK”を入力すれば良い。この方法では、便宜的に“あ行”の子音は‘A’とし、‘ん’や‘ー’は“わ行”に含むことにしている。“盗塁”と入力する場合は“TARA”と入力する。

この方法は2種類に分類できる。T9のように子音の列をひらがなの列に変換した後に普通のかな漢字変換を行う2段階の手法と、Touch me key 10のように子音の列からかな漢字混じりの文に直接変換を行う1段階の手法である。子音のみの入力では候補数が多くなってしまうことが問題であるが、単語の使用頻度や、単語と単語の接続頻度などから、候補を予測することで、ある程度候補数を絞ることが可能である。

#### 2.4.3 ペンによる日本語入力

我々の研究目標は、ペンによる統一的なインタフェースでのコンピュータ操作のために、フローメニューにおいて日本語入力を実現することである。

先に紹介した Quikwriting で日本語を入力することを想定した場合、Quikwriting は漢字変換を考慮した設計になっていないという大きな問題点を持つ。Quikwriting でローマ字を入力できても、入力したローマ字からかな漢字の文章に変換するためのインタフェースをフローメニューとは別に用意する必要がある。

我々はフローメニュー上で日本語入力を実現するために、新たなインタフェースを考えた。次章ではペンによる日本語入力手法の提案と、その実装方法について述べる。

入力デバイス	入力方法	特徴
キーボード	ローマ字入力	もっともポピュラーな方式, かな漢字変換を伴う.
	かな入力 (キータイプ 1 回)	変換操作はローマ字入力と同じだが, 1 回のキータイプで 1 文字打てるため打鍵速度が上がる.
	かな入力 (キータイプ 2 回) [森田 98] [富樫 99a] [増田]	かなを 2 回のキータイプで入力するために工夫された配列.
	かな漢字直接入力 [森田 97] [増村 97] [富樫 99b]	かなと漢字を 2 回以上のキータイプで直接入力するための方法. 1000 文字以上の漢字を覚える必要がある.
ペン	文字認識	もっとも自然な入力方法. 画面に書いた筆跡から文字を認識する. 変換は必要ないが, 決められた枠の中に楷書で書く必要があり, ストローク数も増えて遅くなる.
	ソフトウェアキーボード (SK)	キーボードを画面上でエミュレートする. ペンの移動が増え, キーのボタンを目で追うのが大変. 変換が必要.
	SK + 予測入力 [増井 97]	全体でのキーの押下数は減って, SK 単体よりも速度が向上.
携帯電話	携帯電話の標準入力	携帯電話で広く使われている方法. ‘お’ を入力するには ‘1’ を 5 回押す. 変換を伴う
	携帯電話のポケベル入力	ポケベルで使われていた入力方法で, 2 回の打鍵で入力する. ‘お’ を入力するには ‘1’, ‘5’ と押す. 変換を伴う.
	標準入力 + 予測入力 [増井 97] [TOS]	最近増えてきた方法で, 入力から単語を予測する. 変換は候補の選択で行う.
	子音入力 [T9]	子音の列からひらがなの単語を推測する. 変換が必要.
	子音入力 + 予測入力 [TIYM01]	子音の列から漢字かな混じりの単語を推測する. 変換は候補の選択で行う.

表 2.3: 日本語入力の方法とその特徴

## 第3章 日本語入力システム Popie

### 3.1 本章の概要

第2章で述べたように、フローメニュー上における文字入力には Quikwriting という方法があるが、日本語入力には対応していなかった。我々は、フローメニューにおけるペン型デバイスをを用いた日本語入力手法を提案し、日本語入力システム Popie ( Predictive Operation on PIE shaped menu ) として実装した。

我々の作ったシステムではペンを操作面に触れてからその場にしばらく留める、いわゆるタップ&ホールドによりフローメニューを表示するようにした。タップ&ホールドによるトリガを用いれば、ペンにボタンが付いていない環境でもフローメニューを使用することが可能である。

### 3.2 Popie の特徴

我々の実装した日本語入力システム Popie の特徴は以下の点である。

- 入力も候補の選択もフローメニュー上で行える
- 入力は子音によって行い、フローメニューの特徴を生かし、素早く入力できる
- ユーザによる入力から単語の候補を予測して変換候補として提示する
- 必要に応じて母音の選択が行える

Popie では、フローメニュー上で子音の入力と候補の選択の両方を行うことができることが一番大きな特徴である。ソフトウェアキーボードを使った入力では、ユーザはボタンの位置を目で追いかける必要があったが、Popie では子音による入力をフローメニュー上で行うので、フローメニューにおける10個の子音のメニューを位置で覚えることで、メニューを見なくても子音の素早い入力が可能になる。

また、システムが予測して提示する候補の選択も、フローメニュー上で行うので、子音の入力と候補の選択の操作を、シームレスに滑らかに行うことができる。

### 3.3 子音入力における変換候補

子音のみの入力において、予測や推測を用いることでユーザが意図する候補にほぼ絞込こむことが出来るのは確かであるが、表 3.1 に示すように、入力文字数が少ないと、候補数が爆発的に多く、子音の組み合わせによっては予測による候補の絞込みが難しくなってしまうこ

とがある。我々はこの問題に対して、新たに“母音選択”のインタフェースを用意し、未知語や普段あまり使わないような単語を入力しようとして、子音の列のみの入力ではユーザが意図した候補が表示されなかった場合に、ユーザが必要に応じて母音を選択し、候補数を容易に削減できるようにした。

文字数	組み合わせ数	候補が1以上ある子音組み合わせ数	平均候補数	標準偏差	最大値
1	$10^1 = 10$	10	326.50	249.20	850
2	$10^2 = 100$	99	123.62	127.26	650
3	$10^3 = 1000$	881	21.60	30.83	419
4	$10^4 = 10000$	4886	3.12	8.01	150
5	$10^5 = 100000$	8446	0.18	1.31	96

表 3.1: 子音の文字数とそれにマッチする候補の数

この表は、文字数が  $n$  で構成され得る子音列の全ての組み合わせに対して、子音列の長さや候補の読みの長さが一致する候補がそれぞれいくつあるかを検索し、その平均候補数、標準偏差、最大値を示している。この表の計算には、日本語入力プログラム SKK [SKK] の辞書 SKK-JISYO.L を使用した。

### 3.4 Popie のインタフェース

実装した入力システムのインタフェースを図 3.1 に示す。

インタフェースは入力選択部と候補表示部の2つで構成される。入力選択部では子音の入力、母音の指定、バックスペースや改行などの入力、および入力された文字列から予測される候補の選択を行う。候補表示部では予測によって絞り込まれた候補を表示し、候補の選択時にはフローメニューと色を合わせて、候補を選択しやすくしている。

上部に表示された赤く塗りつぶされた文字列はユーザが入力した子音の列である。この図において、ユーザは“卒業”という単語を入力しようとして“STKYA”という子音の列を入力している。

本システムの子音入力では“AKSTNHMYRW”の10個のキーのみを用いる。濁点や半濁点の有無や、拗音であるかどうかなどは考慮せず、何行の文字であるかということのみを必要とする。ここで、便宜的に“あ行”の母音を‘A’としているため、“そつぎょう”の最後の‘う’に対応する文字が‘A’になっている。また、‘ん’や‘ー’は“わ行”に割り当てた。

#### 3.4.1 子音の入力

子音の入力はフローメニューにおける、メニュー選択と同じ要領で行う。図 3.2 に“田中”と入力するために“TNK”と子音を入力する例を示す。

まず‘T’の入力は、レストエリアから右上の“STN”とラベルがついたオクタントにペンを移動し、そのオクタントから再びレストエリアにペンを戻すことで行う。次に‘N’の入力は、レストエリアから同じく右上の“STN”のオクタントにペンを移動させ、その後‘N’のオ

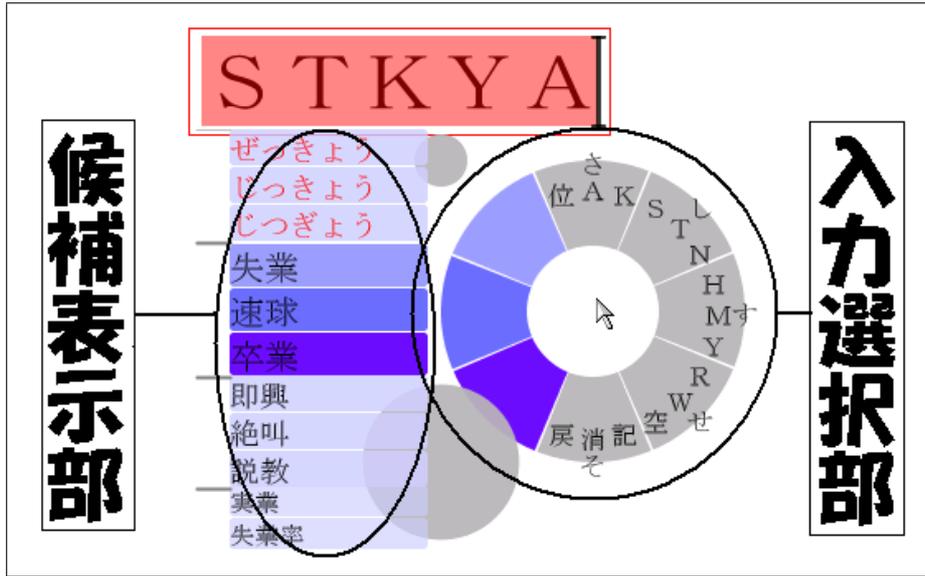


図 3.1: Popie のインタフェース

クタントまでペンを移動した後、レストエリアにペンを戻すことで行う。同様に‘K’の入力は上方向の“AK”のオクタントにペンを移動し、右上の‘K’のオクタントに移った後、レストエリアにペンを戻すことで行う。これら‘T’、‘N’、‘K’の入力はペンを画面から離すことなく一連のストロークで行うことができる。

また、下方向の“戻消記”と表示されたオクタントのメニューを選択し、‘消’メニューを選択することで、文字を1文字消すことが可能である。他に、‘空’メニューでスペース、“タブ”メニューでタブを、‘改’メニューで改行を入力することができる。

### 3.4.2 母音の指定

母音の指定は先頭から1文字ずつ行うものとし、オクタントの更に外側にペンを移動することで指定できるようにした。図 3.3 に母音の指定の例を示す。ユーザが“コイン”と入力しようとして、“KAW”と入力したが、候補には“議員”や“気温”などの候補が挙がったが、目的の“コイン”の候補が見当たらなかったとする。ここでユーザは“こ AW”と、最初の文字の母音を‘お’に指定するために“戻消記”のオクタントにペンを移動した後、そのオクタントの更に外側までペンを移動する。ここで“コイン”という候補が出てきたのでユーザはそれを選択する。

同様にして右側5つのオクタントの外側にペンを移動することで、“あいうえお”の母音を選択することができる。また、選択した文字に濁音、半濁音、あるいは拗音、撥音がある場合は、オクタントからの外側にペンを移動させて母音を選択した後、一度オクタントの中にペンを戻し、目的の文字がラベル付けされたオクタントから再び外側にペンを移動させることで、その文字の選択が可能である。図 3.4 に撥音‘っ’のみを入力する例を示す。

それぞれの状態において選択できる文字は、オクタントとその外側とのペンが横切るべき境界線の上に表示する。また、母音を指定したのち、外側からオクタントに入り、ただちにレ

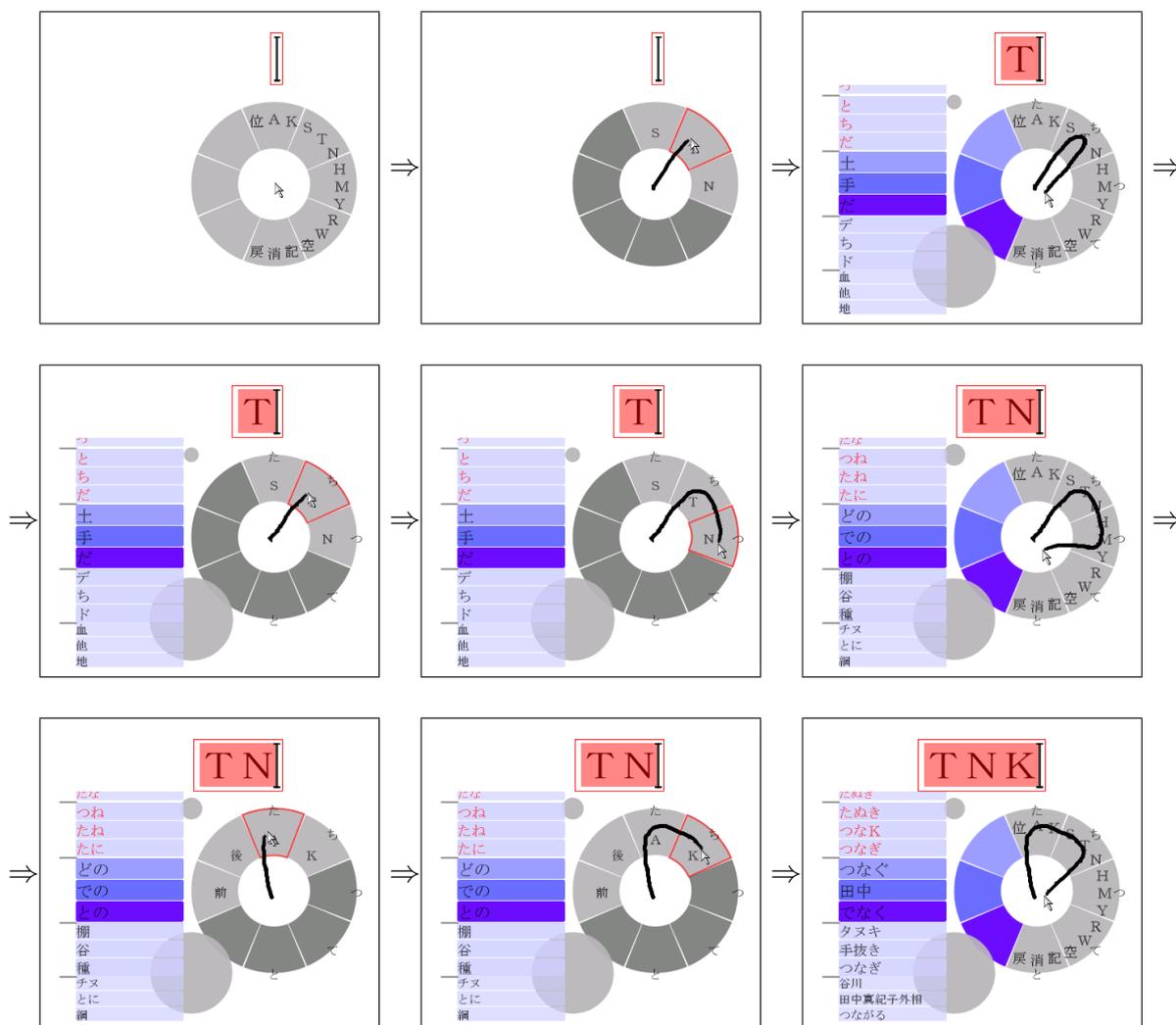


図 3.2: “TNK” の入力例

ストエリアに戻った場合には何も動作しないようにしている。

### 3.4.3 候補の表示と選択

Popie での子音入力候補の変換は子音の列から直接漢字に変換する 1 段階の方法 [TIYM01] と、子音の列からひらがなの読み、漢字と 2 段階に変換する方法 [T9] の両方の候補の変換が行えるようにした。

候補の表示は、予測変換候補および、絞り込み変換候補からなり、それぞれの候補は、黒色と赤色の 2 種類で表示され、黒色は確定、赤色は未確定のまま変換することを示す。

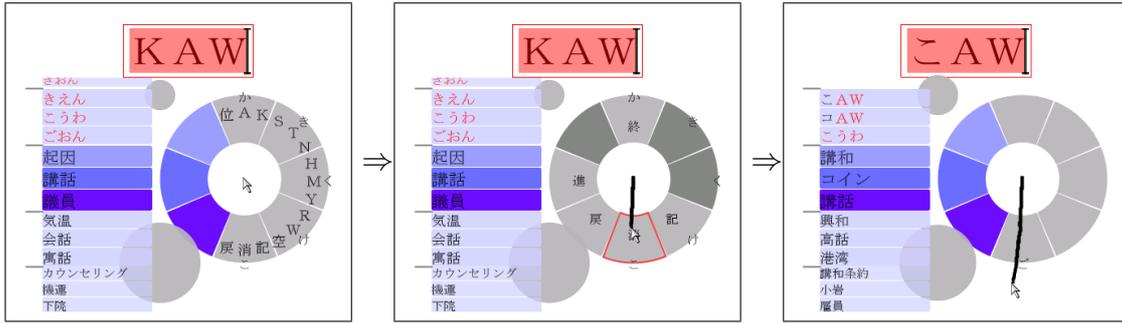


図 3.3: 母音‘あ’の選択例

### 予測変換候補

単語辞書やユーザ学習辞書からの予測や、直前の文字列からユーザが次に入力すると予測される変換候補を黒色で表示する。選択すると文字列が決定される。ユーザの入力した子音の列の長さよりも、読みの短い候補しか見つからなかった場合、途中までの変換候補を黒色で、残りの子音、もしくはひらがなの列を赤色として表示する。この候補を選択すると黒色の部分が確定し、引き続いて赤色の未確定の変換が行われる。

### 絞込み変換候補

辞書からの予測で挙がった候補群の読みのうち、ユーザの入力した子音列の長さまでの読みが同じである候補のスコアの合計が高い読みを抽出し、ひらがなへの未確定変換候補として赤色で読みを表示する。“TNK”と入力したときは赤色の候補“たなか”、“たぬき”などが表示される。ユーザが赤色の絞込み変換候補“たなか”を選択すると、選択された“たなか”に対する変換候補が新たに表示される。

### 候補の選択方法

候補の選択方法は、入力選択部の左3つの青い色のオクタントを用いる。候補を選択する場合これら3つのオクタントのいずれかにペンを移動させ、選択したい候補と同じ色のオクタントからレストエリアにペン戻すことで行う(図 3.5)。

3つのオクタントは上から順に色が濃くなり、対応する候補との色の対応づけを行うことで、ユーザが候補を選択しやすいようにした。

候補の数がどれだけあるかという情報は、候補選択部分の上下に候補数に応じた大きさの円を描くことで表現する。より下位の候補を選択するためには左側3つのオクタントにペンを移動させたあとで、‘ ’や‘ ’が表示される上下のオクタントにペンを移動することで目的の単語が表示されるまでスクロールさせる。ここではオクタントからレストエリアに戻らずにスクロールが実行されるので、上下のオクタントにその左側のオクタントから入り出すことで連続的にスクロールできる。また、上下のオクタントにペンを留めることである一定時間において自動的にスクロールされるようになっている。

スクロールは単語3つ単位で上下にスクロールさせている。表示されている候補の左側に

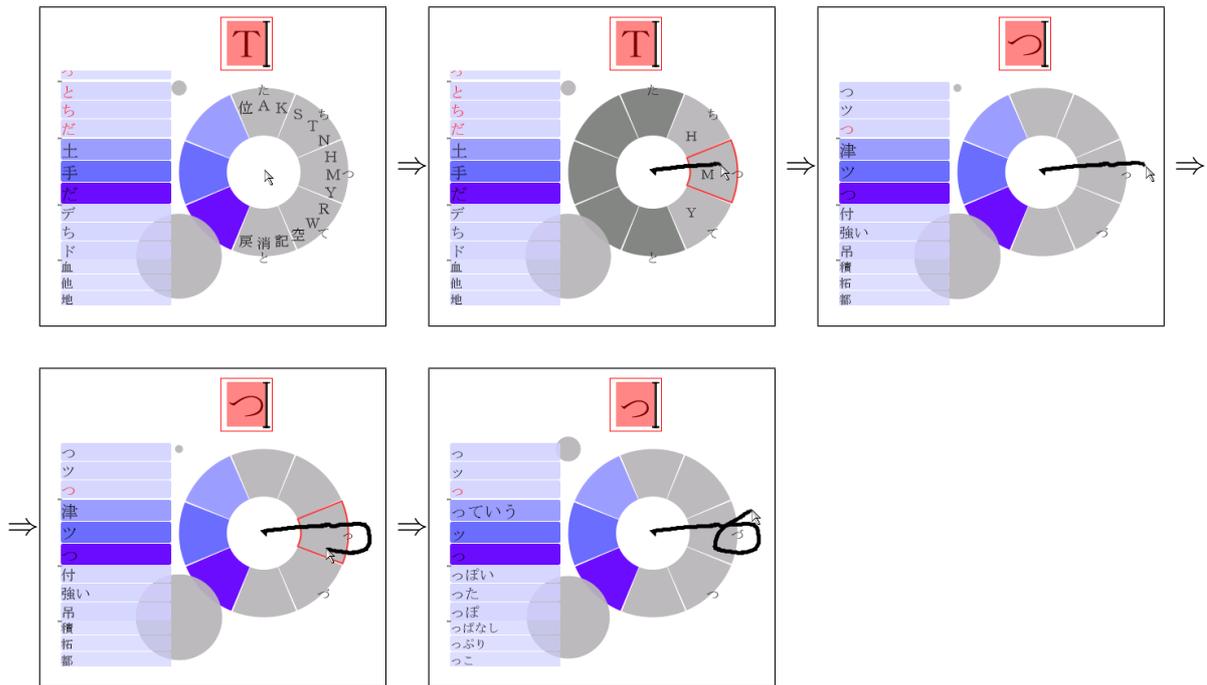


図 3.4: 撥音‘っ’の選択例

引いてある線は3つごとの単語のグループを分かりやすくするためのものである。スクロールして候補を選択する例を図 3.6 に示す。この例においてユーザは“牡蠣”を入力しようとしている。

また、スクロールしたが候補がなく、母音を指定して入力しようと思った場合はさらに右側の灰色になったオクタントからレストエリアに戻ることで、候補を選択せずに元に戻る事ができる。

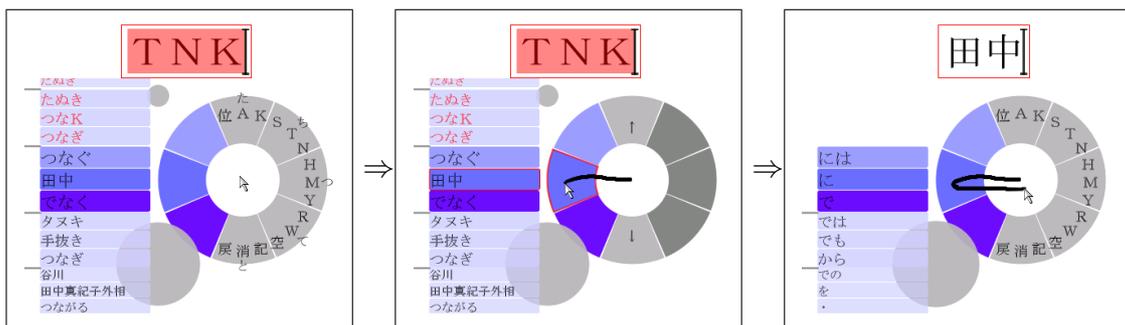


図 3.5: 候補の選択の例

### 3.4.4 キャレット位置の調節

候補の中に思ったような単語がなかった場合に、先に述べた母音選択をすることによっても候補を絞ることができた。Popie ではさらに、キャレットの位置を調整して、文節の長さ

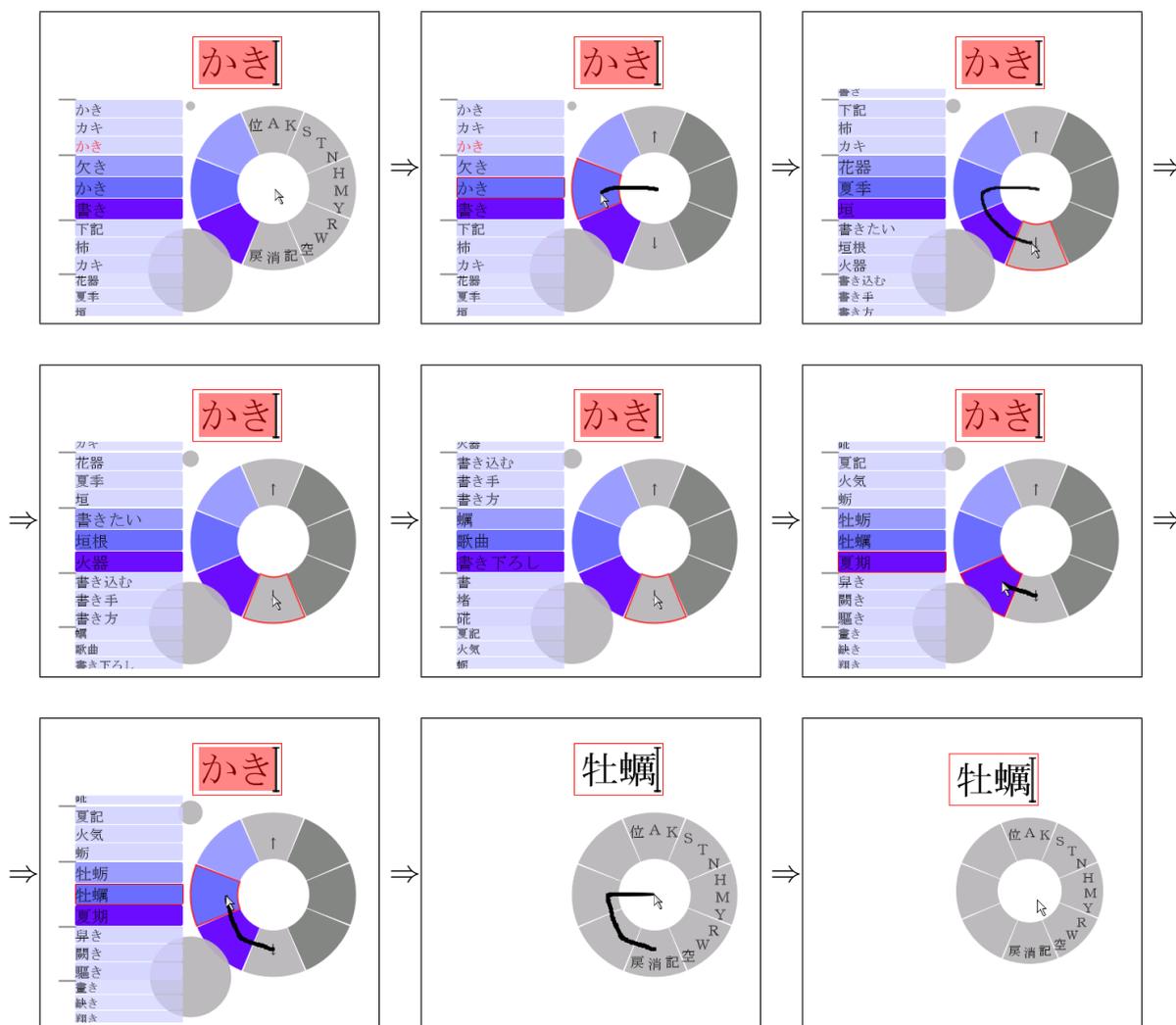


図 3.6: 候補スクロールして“牡蠣”を選択する例

を明示的に指定することで、変換を補助することが可能である。文節の長さの調節は赤色で表示された未確定候補が有る場合のみ行うことができ、未確定文字列が無い場合は同様の操作により、カーレットの位置を変更し、編集する位置を変更することができる。

図 3.7 にカーレット位置の変更をする例を示す。Popie 初期状態における上方向のオクタントは“位 AK”とラベル付けされているが、上方向のオクタントに移動した後、その左側の‘前’、‘後’とラベル付けされたオクタントからレストエリアに戻ることににより、カーレットの位置を調節することができる。

### 3.4.5 アンドゥー・リドゥー

Popie により日本語を編集集中にユーザが行った操作は全て記憶してあり、無限アンドゥー・リドゥー操作が可能である。

図 3.8 にアンドゥーの例を示す。Popie の初期状態における下方向のオクタントは“戻消

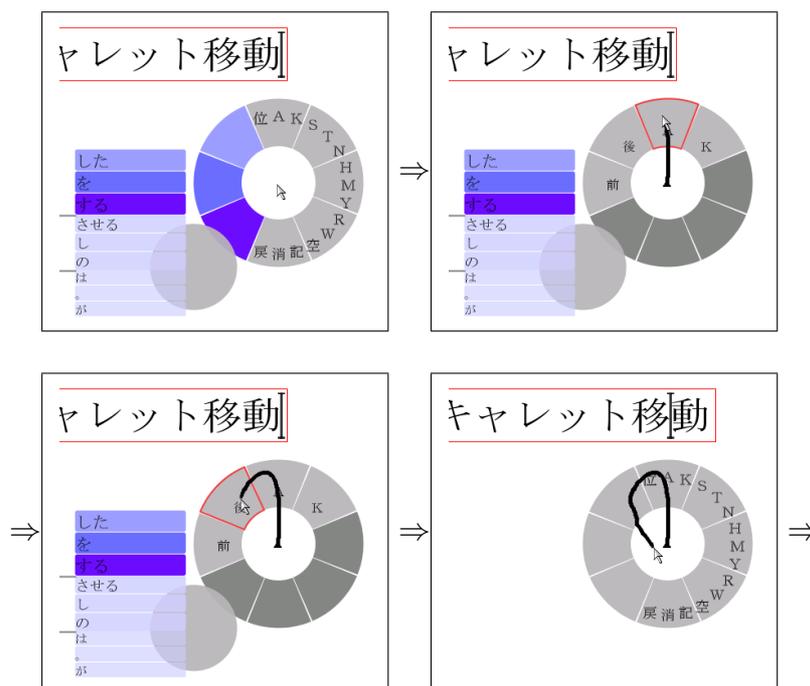


図 3.7: キャレット位置の変更の例

記”とラベル付けされているが、下方向のオクタントから左側の‘戻’、‘進’のオクタントを選択することによってアンドゥ、リドゥができる。

### 3.4.6 記号や英数字の入力

記号の入力は、“かっこ”など記号の名称を“KTK”と子音で入力するか、“きごう”と入力するために“KKA”と子音を入力することで行える。しかし、“KTK”や“KKA”と入力した場合、記号以外の候補も混ざってしまう可能性が高いので、下方向の“戻消記”のオクタントに割り当てられたメニューから‘記’メニューを選ぶことにより未確定文字列としての“きごう”を簡単に入力することができる。この場合、予測機構はユーザが“きごう”と入力したと想定して予測するので、記号の候補が表示される。

また、英数字を入力するには、まず“いち”や“えー”に対応する子音を入力し最初の英数字を選択する。その後は予測機能によって候補に真っ先に英数字が表示される。英数字の候補を絞り込むには、先頭の読みを入れることによって簡単に行うことができる。

### 3.4.7 Popie のメニュー配置

Popie のメニューの配置を表 3.2 に示す。縦軸が初期状態のメインメニュー項目の方向を示し、横軸がメインメニューに対するサブメニューである。サブメニューは、メインメニューの位置からあまり離れないように配置した。メインメニューの位置からサブメニューの位置が離れてしまうと、そのメニューの選択に手間がかかり、ユーザがメニューを目で追いかける必要が出てしまうからである。

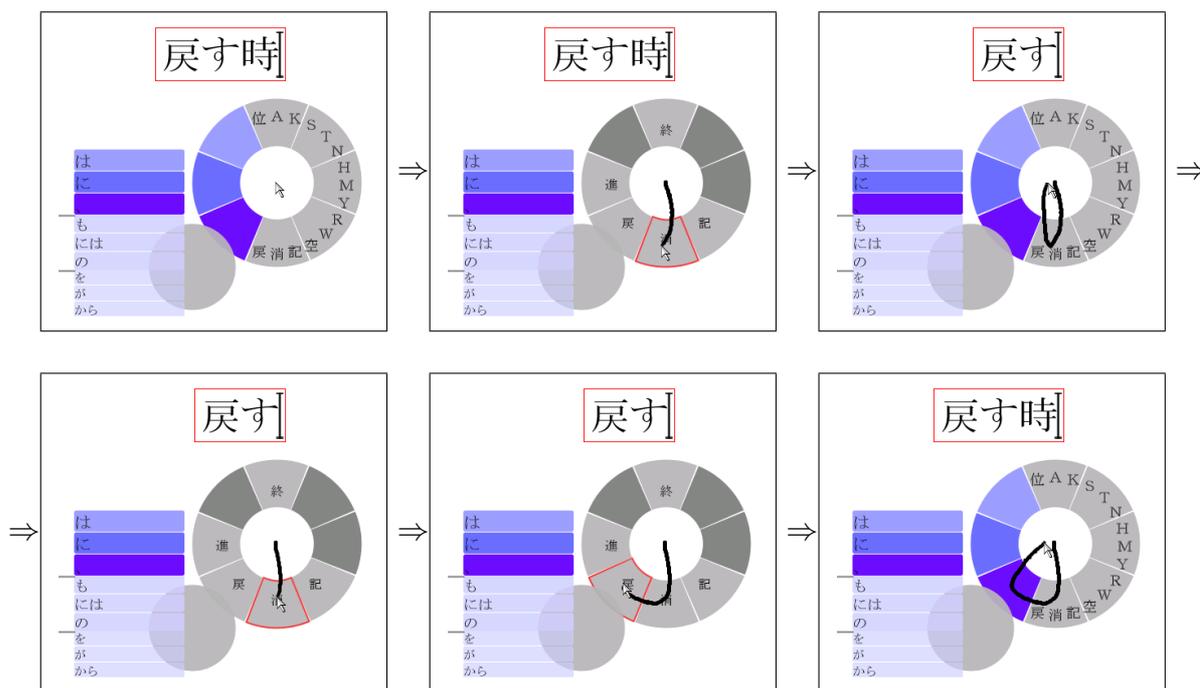


図 3.8: アンドゥ・リドゥーの例

逆に、あまり使わない、もしくは間違ってしまうと困るサブメニューの位置はメインメニューの位置から離すべきである。

		サブメニュー								
		上	右上	右	右下	下	左下	左	左上	
メ イ ン メ ニ ュ ー	上	あ行	か行					キャレット前	キャレット後	
	右上	さ行	た行	な行						
	右		は行	ま行	や行					
	右下			ら行	わ行	空白	改行	タブ		
	下	終了		リドゥ	アンドゥ	消去	記号			
	左下	スク				スク	候補選択			
	左	ロール				ロール				
左上	上				下					

表 3.2: Popie のメニューの配置

## 3.5 Popie の実装

### 3.5.1 辞書と検索

Popie で使用する固定辞書は毎日新聞社の CD-毎日新聞 2001 年度版 [毎日 01] の新聞データをを用いて作成した。

検索する辞書	単語辞書	単語間関係辞書	ユーザ辞書
子音の列の長さと同候補の読みを一致	20000	10000	20000
子音の列と同候補の先頭が一致	5000	20000	30000

表 3.3: 検索方法と検索辞書に対するスコア

新聞のテキストデータは茶筌 [松本 00] により形態素解析し、単語ごとの頻度情報や単語間の前後関係の頻度情報を解析して作成した。単語辞書には単語、読み、頻度が登録されている辞書と、単語、読み、直前に出現頻度の高い単語、出現頻度が登録されている 2 つの辞書を用いた。

また、ユーザが入力する文章をもとにユーザ辞書を作成し、各ユーザに適応するように学習を行う。学習はユーザが入力したすべての文字列を単語、単語間の前後関係の情報として利用する。ユーザがより多く使う単語、単語の列はより候補の上位に表示されるようになる。ユーザが使用して学習された単語は、その後しばらく使わなかったものから順次ユーザ辞書から消去されるように、少しずつ使用頻度を下げよう方法を取った。このことによりユーザ辞書が肥大しすぎる事を防いでいる。ただし、固定辞書にない単語についてはユーザ辞書から消えないようにする。

システムはこれらの辞書を用いて子音の列からの変換候補の検索を行う。それぞれの変換候補は単語の頻度情報、単語間の前後関係の頻度情報、および検索方法に基づくスコアをもち、スコアの高い順にユーザに提示される。検索方法としては、

- 子音の列の長さと同候補の読みを一致させて検索
- 子音の列と同候補の先頭が一致するように検索
- 子音の列の先頭と同候補が一致するように検索し、残りの子音に対して単語を組み合わせて検索

の 3 つである。

それぞれの検索方法によるスコアは現実装では表 3.3 のようになっている。子音の列の長さと同候補の読みを一致させる検索に対してスコアを高くすると、ユーザが入力した子音の長さに合うような候補が上位に表示される。逆に、子音の列と同候補の先頭が一致する検索に対してスコアを高くすると、ユーザの意図しない候補が多く表示される可能性があるが、少ない入力でもより長い単語を入力できるようになる。

本実装では、単語間関係辞書とユーザ辞書に関して、後者のより長い単語を優先的に上位の候補となるような設定にした。これは、ユーザ辞書が、ユーザからの入力の学習によって作られた辞書であるため、ユーザの良く入力する単語を多く含む可能性があり、また、単語間関係辞書は、ある単語とある単語の結びつく頻度を収めた辞書であるため、ユーザの入力よりも読みの長い単語であっても、ユーザが意図した単語である可能性が高いからである。

逆に単語辞書からは候補の読みと長さを一致する検索を優先させ、より確実に変換を行えるようにした。

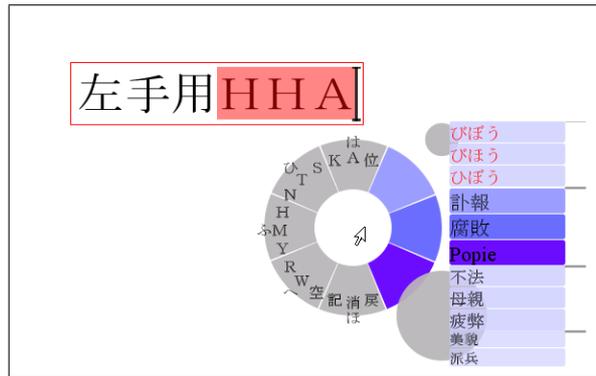


図 3.9: 左利き用 Popie インターフェース

### 3.5.2 左利き

本章で説明した Popie は右利きのユーザを想定している。左利きの利用者には左右が反対になるものを用意した（図 3.9）。これはペン型デバイスを使用すると操作領域をユーザ自身の腕が隠してしまうためである。子音入力や母音選択の部分は覚えてしまえば隠れてしまってもさほど問題がないが、表示位置が変化する候補選択の部分は隠れてしまうと操作しづらいので、ユーザ自身の腕で隠れないような配置にした。

## 第4章 システムの実装

### 4.1 実装環境と言語

システムの実装に使用した言語は Java で, Java™2 SDK, Standard Edition バージョン 1.4.1.01 を使用した. システム全体ではおよそ 12000 行あるが, 本論文で提案する日本語入力 Popie に必要なプログラムはおよそ 5000 行である.

### 4.2 クラス階層

システム全体のクラス図を 4.1 に示す. 本システムは大きく分けてオブジェクトのクラス群, フローメニューのクラス群, アンドゥ・リドゥのクラス群, 日本語処理のクラス群からなる. 本論文で紹介した Popie は主にフローメニューと日本語処理のクラス群からなり, 文字列の表示はオブジェクトのクラス群が担当している.

#### 4.2.1 オブジェクトのクラス群

このクラス群はオブジェクトの表示や追加, 削除, ファイルへのセーブやファイルからのロードを担当するクラス群である.

“FlowMain” クラスはトップレベルのウィンドウで, “FlowPanel” クラスを持つ. “FlowMain” クラスは主に初期化やコマンドラインからの引数を解釈するクラスである.

“FlowPanel” クラスは “Page” クラスと “BackGround” クラスのリストを保持し, ページの追加や削除などの操作を管理している.

“Page” クラスはオブジェクトのリストを保持し, オブジェクトの追加, 削除, 描画等を管理するクラスである. それぞれの “Page” クラスは “UndoRedo” クラスを保持し, このことでページ単位のアンドゥ・リドゥを可能にする. また “Page” クラスを継承した “BackGround” クラスを 1 つ参照し, 複数のページで同一のバックグラウンドを表示することができる. “BackGround” クラスは “Page” クラスと同様にオブジェクトを保持し “Page” クラスと同様に編集可能である.

表示されるオブジェクトは, 全て抽象クラス “Obj” を継承し, 回転やズーム, 色の変更といった操作を行えるインタフェースを持つ. “Obj” クラスを継承したクラスには “Circle” や “Rectangle”, “Str” 等がある. 本論文で紹介したシステムでは “Str” クラスを使用して文字列の表示を行っている. また “Obj” クラスはそれぞれ “toString()” 関数と “parse()” 関数を持っており, ファイルにセーブする際には “toString()” 関数を用いて属性値等を文字列として出力し, 逆にファイルからロードする際には “parse()” 関数により文字列からオブジェクトを再構成することができる. 上記 2 つの関数のカスタマイズにより, CSV などの一般的な

ファイル形式に保存することも可能である。

#### 4.2.2 フローメニューのクラス群

“FlowMenu” クラスはフローメニューを実現するためのクラスで、状態の管理とオーナークラスである “FlowPanel” クラスとのやりとりのみを行う。フローメニューの描画やアクションの操作等の実質的な処理は全て抽象クラス “State” の実装が行うようになっている。

抽象クラス “State” の実装にはいくつかのアダプタクラスがある。それぞれのアダプタクラスはフローメニューのいくつかの状態において共通化できる描画や、ペンがメニュー上のどの領域に入っているかを特定するメソッドを持っている。フローメニューを表示したときの状態である “DefaultState” クラスや、編集のためのメニューを持った状態である “EditState” クラスは “StateAdapter” クラスを継承している。回転のための状態である “RotateState” クラスやズームのための状態 “ZoomState” クラスはジョグダイアル状のメニューであり、“JogAdapter” クラスを継承する。この他に色を選択するための “ColorJogAdapter” クラスがあり、色を決定するための状態がこれを継承する。また、Popie での操作を処理する状態は “PopieAdapter” クラスを継承する。Popie の初期状態である “PopieState” クラス、オクタントの 8 状態をまとめた “PopieRingState” クラス、母音や濁音、撥音、拗音などの選択を行う状態 “PopieOut” クラスである。

#### 4.2.3 日本語処理のクラス群

Popie のそれぞれの状態は “StringStateManager” クラスに行った操作を “Keyboard” クラスのキーとして送信する。キーには子音の ‘A’, ‘K’, ‘S’ や、母音の ‘a’, ‘i’, その他アンドウ・リドゥのためのキーや候補選択のキー等がある。操作をキーとして扱うことで、操作するインタフェースが変わったとしても実装の変更が容易に行える。

“StringStateManager” クラスは、次々に来るキーの列を処理して、文字列の状態である “StringState” クラスの状態を変化させたり、“Cand” クラスにより候補を生成したりする。

“Cand” クラスは辞書クラス “Dictionary” の単語から候補を抽出し、候補やその読みを持った “CandKey” クラスのリストを生成する。“Dictionary” クラスは単語や読み、および直前に来る単語などの情報を持った “Word” クラスのハッシュを持つ。“Word” クラスのインスタンスはそれぞれの読みの先頭部分でハッシングしており、“Cand” クラスは欲しい単語の読みを渡すことで、“Word” クラスのインスタンスのリストを取得できる。

“Str” クラスは “StringState” クラスを持っており “StringStateManager” によって変更された文字列情報を画面に表示している。文字列情報には未確定候補の文字やキャレットの位置などがある。

#### 4.2.4 アンドウ・リドゥのクラス群

“UndoRedo” はアンドウとリドゥを実現するためのクラスである。ユーザが行った操作は全て “UndoRedo” クラスに抽象クラス “Command” の実装クラスとして登録される。“Command” の実装クラスには “Rotate” や “Move”, “Typing” などがあり、これらは操作対象で

あるオブジェクトへの参照と操作を実行する直前と直後のオブジェクトの属性などを保持する。

“UndoRedo” クラスに対して “undo()” もしくは “redo()” 関数が呼ばれると, “UndoRedo” クラスは現在の位置にあるコマンドに対して “undo()” もしくは “redo()” を実行する. このとき “undo()” や “redo()” を実行したというコマンドも記録され, 全ての操作履歴が保存されるようになっているが, 一度アンドゥで操作を戻ってから別の操作をしたときには, 過去に行った操作は再現されないような実装にした.

それぞれの “Str” クラスも “Page” クラスと同様に “UndoRedo” クラスを持ち, オブジェクトの操作とは別に文字列の編集のアンドゥ, リドゥが行えるようになっている.

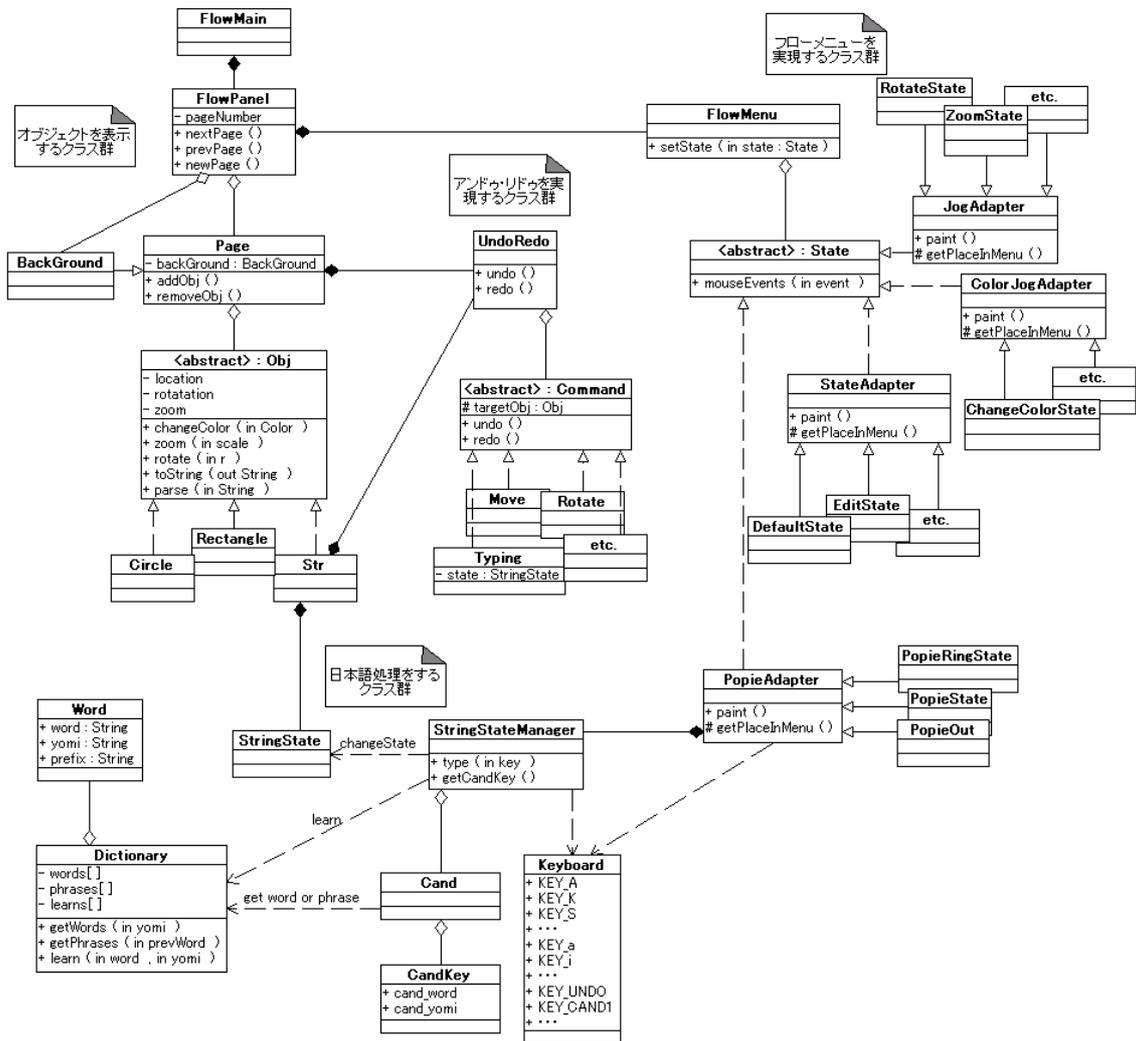


図 4.1: システムのクラス図

## 第5章 システムの評価実験

### 5.1 実験方法

#### 5.1.1 被験者

被験者は20代前半から30代前半までの男性4人を対象に行った。被験者は普段からキーボードの操作には十分に慣れていたが、Popieの操作の慣れには個人差があった。

#### 5.1.2 方法

100文字程度の文章3種類について、以下の入力方法でそれぞれで3回ずつ入力を行う。

1. キーボードとIMEを使って入力する方法 [Keyboard]
2. タブレットPCの文字認識機能により手書き入力をする方法 [Pen]
3. Popieにより入力する方法 [Popie]

Popieは学習機能があるので、それぞれの被験者が実験を始める段階で、学習辞書をリセットして実験を行った。

#### 5.1.3 実験環境

実験はTOSHIBA Dynabook SS 3500, WindowsXP TabletPC Edition, Mobile PentiumIII 1.33Ghz 512MBを用いて行った。

#### 5.1.4 入力する文章

文章1 単語辞書は単語を使用頻度順に並べたもので、図2のような読みと単語を並べた単純なテキストファイルです。ユーザーが次に入力する単語の予測に現在のコンテキストを利用できない場合は、この単語辞書が使用されます。(101文字)

Unix Magazine, Vol.13, No. 5, 1998. インターフェイスの街角(6) - POBoxの予測手法と辞書の作成より抜粋

文章2 帰還直前に空中分解して7人の乗組員が死亡した米スペースシャトル・コロンビアの事故で、米航空宇宙局(NASA)は4日、機体に異変が起き始めたところに落下した破片を西部の2州で見つけた、と発表した。(96文字)

asahi.com 2003 年 02 月 05 日記事「翼の一部？ 米西部 2 州で破片発見 重要証拠の可能性」より抜粋

文章 3 力士の体に詳しい医師が、けがの現状や筋力、体脂肪とけがの因果関係を報告。また、ドーピング（不正薬物使用）がけがの一因と指摘されている問題では、専門家が薬物の危険性や副作用などを指摘した。（93 文字）

Yomiuri On-Line 2003 年 02 月 05 日記事「力士のけが予防講座開催」より抜粋

## 5.2 実験結果

表 5.1 はそれぞれの文章をそれぞれの入力方法で入力したときの時間を秒であらわしている。図 5.1、図 5.2、図 5.3、図 5.4 は、被験者ごとに、グラフにしたものである。グラフの横軸は入力の回数で、縦軸は入力にかかった時間（秒）である。

		文章 1			文章 2			文章 3		
		1	2	3	1	2	3	1	2	3
被験者 A	Keyboard	60	51	48	69	58	58	72	59	59
	Pen	211	257	178	203	212	202	205	233	187
	Popie	648	204	121	559	201	109	563	273	132
被験者 B	Keyboard	52	47	47	56	57	53	58	46	52
	Pen	182	186	178	176	154	164	195	151	180
	Popie	240	121	86	253	116	97	224	125	117
被験者 C	Keyboard	102	99	76	105	78	73	90	76	82
	Pen	213	222	211	195	184	203	219	214	244
	Popie	698	322	156	612	213	151	482	240	194
被験者 D	Keyboard	89	66	74	89	72	87	80	81	70
	Pen	247	267	217	220	187	209	225	207	174
	Popie	434	393	322	197	152	165	126	122	137

表 5.1: それぞれの入力方法における入力時間（単位 秒）

## 5.3 評価

被験者全員がほぼ同じような傾向を示すグラフになった。キーボードと手書きによる入力は普段から良く行う操作であり、3 回繰り返しての入力でも、入力速度に変化はなかった。

Popie での入力は、各文章の 1 回目の入力は、手書き入力よりも少し時間がかかったが、2 回目以降は全員が手書き入力と同じくらいか、それよりも速く入力できていた。これは Popie の学習機能によるものだと考えられる。ユーザが日常的に Popie を使えば、安定して手書きよりも速い速度で入力が可能になるだろう。

手書きでは全員が 30 文字 / 分前後の入力速度しか出ていないが、Popie では 60 文字 / 分近い入力速度を出している被験者もいた。

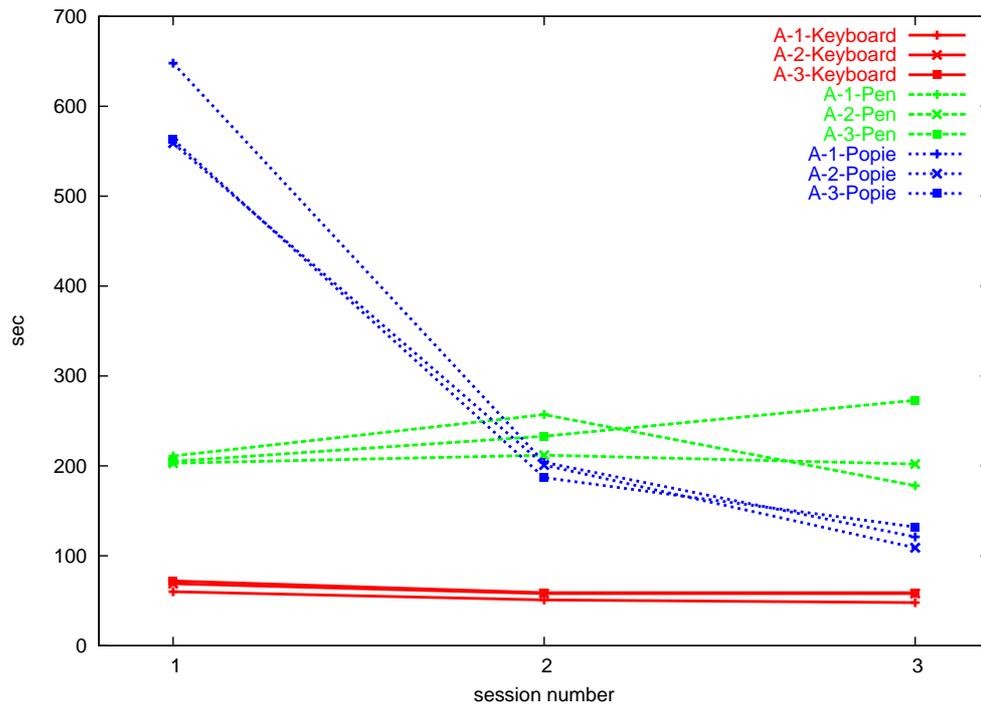


図 5.1: 被験者 A の入力時間のグラフ

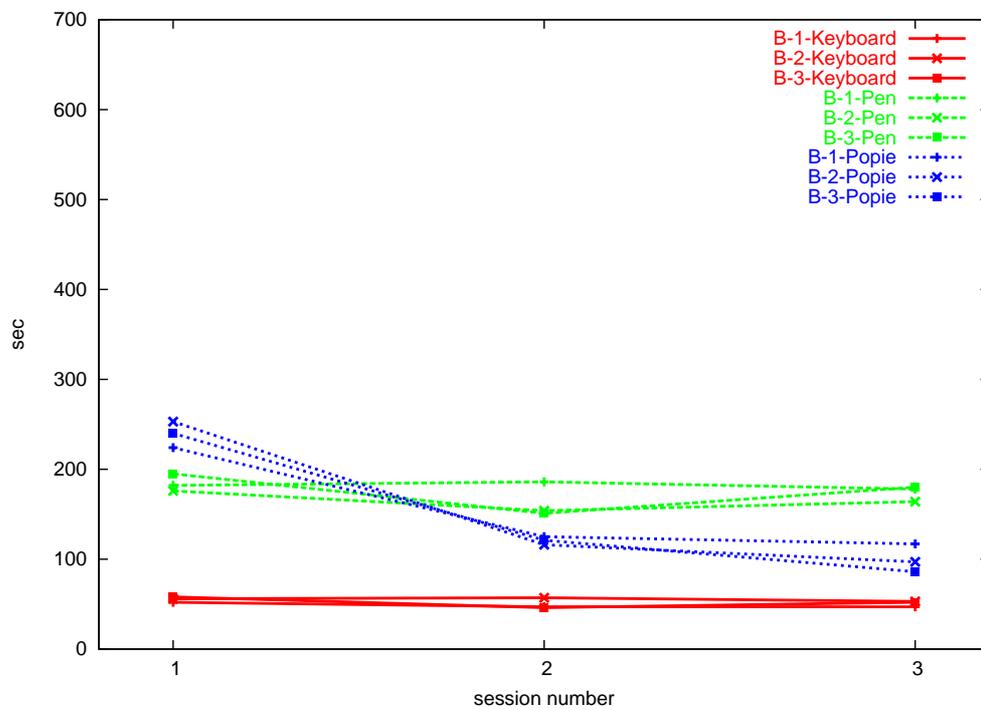


図 5.2: 被験者 B の入力時間のグラフ

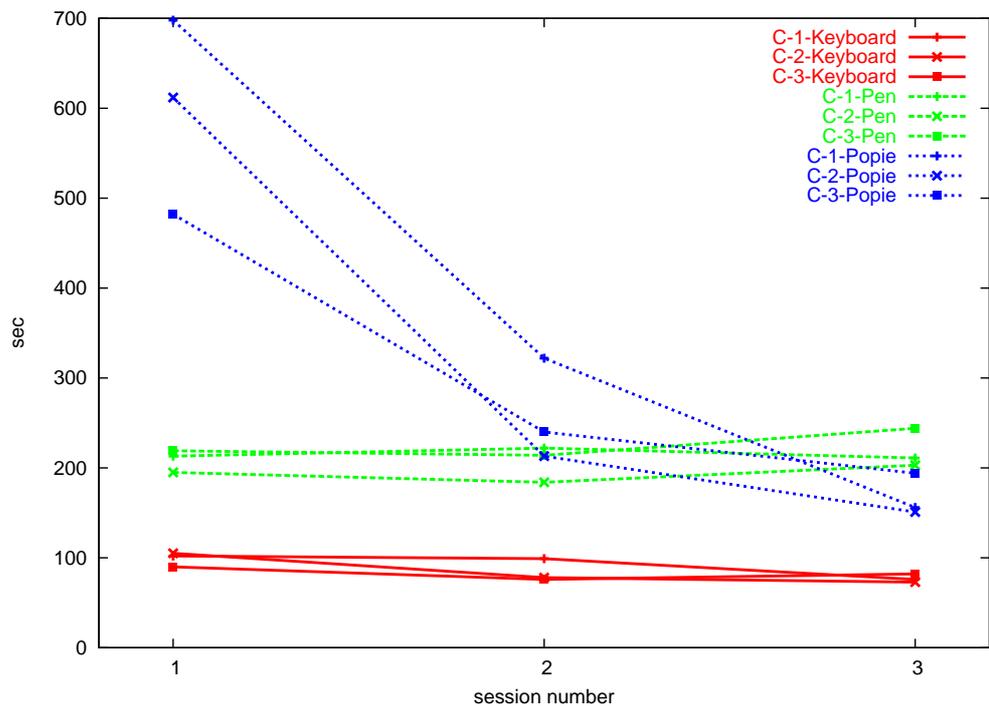


図 5.3: 被験者 C の入力時間のグラフ

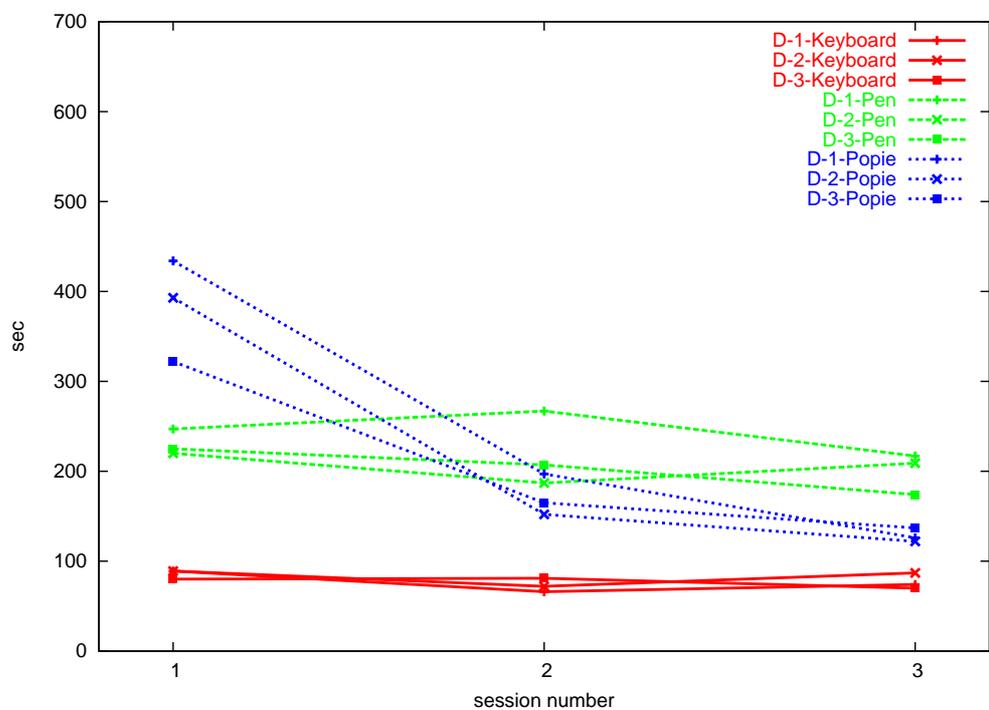


図 5.4: 被験者 D の入力時間のグラフ

この実験の結果より、Popie での入力の手書き入力よりも速く、実用的にも問題なく使用できると考えられる。

## 第6章 まとめ

本研究ではフローメニュー上での日本語入力インタフェース Popie を提案し、実装した。Popie ではフローメニューのメニュー選択をすることで、子音の入力と予測候補の選択をシームレスに行うことができる。また、子音が表示されるメニュー項目の位置を覚えることで、素早い子音の入力が可能になった。

実験結果より、Popie での日本語入力は実用的な入力速度がある事が確認され、これにより、日本語においてもペン型デバイスによる統一的な操作が可能になった。

## 謝 辞

本研究を進めるにあたり、指導教官である田中二郎教授、ならびに助手の志築文太郎先生、三浦元喜先生からは、大変貴重な助言、ご指導をいただきました。心より感謝致します。

また、田中研究室の皆様にも、ゼミ等を通じて多くの意見を頂きました。特に Web グループの糸賀裕弥さん、椿山智弘さん、小野田紗智さんには多くの意見を頂きました。ありがとうございました。

最後に、自分を支えてくれた両親やすべての友人に心より感謝致します。

## 参 考 文 献

- [Bli95] Conrad H. Blickenstorfer. Graffiti: Wow! *Pen Computing Magazine*, pp. 30–31, January 1995.
- [EXi] EXideas. MessagEase. <http://www.exideas.com>.
- [GR93] David Goldberg and Cate Richardson. Touch-typing with a stylus. In *Proceedings of ACM INTERCHI 1993 Conference on Human Factors in Computing Systems (CHI 1993)*, pp. 80–87, April 1993.
- [GW00] Francois Guimbretiere and Terry Winograd. FlowMenu: Combining command, text, and data entry. In *Proceedings of ACM User Interface Software and Technology 2000 (UIST 2000)*, pp. 213–216, May 2000.
- [MA98] Jennifer Mankoff and Gregory D. Abowd. Cirrin: A word-level unistroke keyboard for pen input. In *Proceedings of User Interface Software and Technology 1998 (UIST 1998)*, pp. 213–214, November 1998.
- [MZ99] I. S. MacKenzie and S. Z. Zhang. The design and evaluation of a high performance soft keyboard. In *Proceedings of the ACM Conference on Human Factors in Computing Systems 1999 (CHI 1999)*, pp. 25–31, May 1999.
- [Per98] Ken Perlin. Quikwriting: Continuous stylus-based text entry. In *Technical Note of ACM User Interface Software and Technology 1998 (UIST 1998)*, November 1998.
- [SKK] SKK Openlab. SKK. <http://openlab.ring.gr.jp/skk/index-j.html>.
- [Sol98] Textware Solutions. Fitaly. <http://www.fitaly.com>, 1998.
- [T9] T9. T9 text input home page. <http://www.t9.com>.
- [TIYM01] K. Tanaka-Ishii, Y. Inutsuka, and M. Takeichi. Japanese text input system with digits –can japanese text be estimated only from consonants?-. In *Proceedings of Human Language Technology Conference 2001 (HLT 2001)*, March 2001.
- [TOS] TOSHIBA Corporation. Mobile Rupo. <http://www.toshiba.co.jp>.
- [VN94] Dan Venolia and Forrest Neiberg. T-cube: A fast, self-disclosing pen-based alphabet. In *Proceedings of the ACM Conference on Human Factors in Computing Systems 1994 (CHI 1994)*, pp. 265–270, April 1994.

- [ZHS00] Shumin Zhai, Michael Hunter, and Barton A. Smith. The metropolis keyboard - an exploration of quantitative techniques for virtual keyboard design. In *Proceedings of the ACM Symposium on User Interface Software and Technology 2000 (UIST 2000)*, pp. 119–128, November 2000.
- [小松 02] 小松弘幸, 高林哲, 増井俊之. 文書蓄積システム kukura を用いた予測入力. In *Proceedings Workshop on Interactive Systems and software 2002 (WISS 2002)*, pp. 43–47. 日本ソフトウェア科学会, December 2002.
- [松本 00] 松本裕治, 北内啓, 山下達雄, 平野善隆, 松田寛, 高岡一馬, 浅原正幸. 日本語形態素解析システム『茶釜』 version 2.2.1 使用説明書. 奈良先端科学技術大学院大学, December 2000.
- [森 02] 森稔, 澤木美奈子. [prpr サーベイシリーズ] 低品質文字の認識手法とその応用に関するサーベイ. 信学技報, March 2002.
- [森田 97] 森田祐二. TUT-Code. [http://www2b.biglobe.ne.jp/~g\\_morita/tut-code](http://www2b.biglobe.ne.jp/~g_morita/tut-code), 1997.
- [森田 98] 森田正典. M 式キーボード. <http://121ware.com/apinfo/content/mworld/>, 1998.
- [増井 97] 増井俊之. ペンを用いた高速文章入力手法. In *Proceedings of Workshop on Interactive Systems and software 1997 (WISS 1997)*, pp. 51–60. 日本ソフトウェア科学会, 近代科学社, December 1997.
- [増村 97] 増村成吾. G-Code. <http://www.asahi-net.or.jp/~QX5S-MSMR/index.html>, 1997.
- [増田] 増田. チョイ入力. <http://member.nifty.ne.jp/kb/Choi.htm>.
- [富樫 99a] 富樫雅文. 中指シフト方式仮名文字配列 花. [http://member.nifty.ne.jp/togasi/hana\\_no\\_kuni/](http://member.nifty.ne.jp/togasi/hana_no_kuni/), 1999.
- [富樫 99b] 富樫雅文. 超多段シフト方式日本語入力システム 風. <http://member.nifty.ne.jp/togasi/>, 1999.
- [毎日 01] 毎日新聞社. CD-毎日新聞 2001 年度版, 2001.