

筑波大学 情報学群 情報メディア創成学類

卒業研究論文

スマートウォッチの竜頭を用いたかな文字入力 手法

白根 薫

指導教員 志築 文太郎

2023年2月

概要

スマートウォッチにおける文字入力には、ユーザ自身の指よりもターゲットが小さいことによって生じる誤入力が生じる問題、およびユーザ自身の指によってスクリーンが覆われる問題がある。さらに、スマートウォッチのスクリーンが標準キーボードによって占有される問題がある。これらの問題を解決するために、スマートウォッチの竜頭を用いた、かな文字入力手法を実装した。本手法は、ユーザが竜頭を回転および簡単なスクリーン上のタップ操作を行うことによって、かな文字入力を行う。また、キーボードレイアウトを工夫することによってスクリーンオクルージョン問題を解消する。

目次

第1章	序論	1
1.1	背景	1
1.2	目的	1
1.3	アプローチ	2
1.4	貢献	2
1.5	本論文の構成	2
第2章	関連研究	3
2.1	スマートウォッチにおけるタッチパネル上の操作を用いた文字入力手法	3
2.1.1	キーボードを工夫した文字入力手法	3
2.1.2	フリック操作を用いた文字入力手法	3
2.1.3	スワイプジェスチャを用いた文字入力手法	4
2.2	スマートウォッチの竜頭を用いた研究	4
2.2.1	スマートウォッチの竜頭を用いたインタラクション性能の調査	4
2.2.2	スマートウォッチの竜頭を用いた文字入力手法	4
第3章	システム	5
3.1	竜頭を用いた文字入力の方法	5
3.1.1	子音選択	5
3.1.2	母音選択	5
3.2	実装	5
3.2.1	システム環境	5
3.2.2	レイアウト	7
第4章	実験	8
4.1	評価指標	8
4.1.1	文字入力速度 (CPM)	8
4.1.2	エラー率 (CER)	8
4.2	実験手順	8
4.3	結果	9
4.4	考察	9
第5章	議論	10

第 6 章 結論	11
謝辞	12
著者論文リスト	13
参考文献	14

図目次

3.1	子音選択の遷移図	6
3.2	母音選択の遷移図	6
3.3	本システムのレイアウト. 左図：三文字分のスペースにて構成されているレイアウト（レイアウト 1），中央図：1 文字分のスペースにて構成されているレイアウト（レイアウト 2），右図：入力欄のみにて構成されているレイアウト（レイアウト 3）	7

第1章 序論

本論文は、スマートウォッチの搭載されている竜頭を用いたかな文字入力手法を実装することを目的とする。本章では、まず、本研究の背景を述べ、その後目的、アプローチ、貢献、および本論文の構成を述べる。

1.1 背景

近年、スマートフォンおよびスマートウォッチなどのウェアラブルデバイスの需要が高まっている。特にスマートウォッチは、主要な入力モダリティはタッチに依存しているため、小さいスクリーンにてテキスト入力を行うことが困難である。そのため、スマートウォッチのような小さなスクリーン上にてユーザ自身の指よりもターゲットが小さいことによって生じる誤入力が生じる問題、およびユーザ自身の指によってスクリーンが覆われる問題（ファットフィンガ問題[?])がある。さらに、スマートウォッチのスクリーンがQWERTTYキーボード、および日本語キーボードによって占有される問題（スクリーンオクルージョン問題）もある。これらの問題を解決するためにスマートウォッチには、音声入力が搭載されている。しかし、音声入力には、雑音の多い場所および公共の場にて使用することが困難であるという問題がある。

今までに、スマートウォッチに関する文字入力手法の研究が行われた。従来のQWERTTYキーボードを改良し文字入力を行う研究[?], フリック操作を用いた文字入力を行う研究[?], およびスワイプジェスチャを用いた文字入力を行う研究[?]が行われた。これらの研究は、小さいスクリーンにて器用な操作が求められる。

そこで本研究は、スマートウォッチの竜頭を用いて、スクリーン上にて器用な操作を必要とせず、スクリーンオクルージョン問題を解消できるような手法を実装する。

1.2 目的

スクリーンオクルージョンの低いキーボードレイアウトを作成し、スマートウォッチに搭載されている竜頭を上下に回転させる操作および簡単なスクリーン上の操作にてかな文字入力を行う。

1.3 アプローチ

竜頭を上下に回転させ、子音選択および母音選択を行う。そして、スクリーン上のタップ操作を行うことによって、選択したキーを決定する。なお、竜頭の回転によって選択するキーが遷移した際に触覚フィードバックを付与することによって、より正確な操作を実現する。

1.4 貢献

本研究の貢献を以下に示す。

- スクリーンオクルージョンを低減するレイアウトを実装した。
- 竜頭を用いたかな文字入力システムを実装した。
- 本手法の性能評価を行った。

1.5 本論文の構成

本論文の構成を示す。第1章では、本研究の背景、目的、アプローチ、貢献、および構成を示した。第2章では、本研究の関連研究を示す。第3章では、本手法の概要および実装を述べる。第4章では、本手法の入力性能を評価するために行った実験を述べる。第5章では、本手法の課題、および改善点を議論し、今後の課題を示す。第6章では、本研究の結論を述べる。

第2章 関連研究

本手法は，スマートウォッチの竜頭を用いたかな文字入力手法である．したがって本章では，スマートウォッチにおける文字入力手法，およびスマートウォッチの竜頭を用いた文字入力手法を示し，本手法との位置付けを述べる．

2.1 スマートウォッチにおけるタッチパネル上の操作を用いた文字入力手法

従来の文字入力手法は，QWERTY キーボードおよび日本語キーボード上にて，タップ操作およびフリック操作にて文字入力を行う．しかし，従来の手法では小さいスクリーンにおいて，正確に文字を入力することが困難である，およびスクリーンオクルージョン問題がある．そこで，本節では上記の問題を解決するための文字入力手法を示す．

2.1.1 キーボードを工夫した文字入力手法

本節では，QWERTY キーボードを工夫した文字入力手法における研究を示す．ZoomBoard[?] は，キーボードをタップによって，タップした周辺のキーを拡大し，他のキーの誤タップを防ぐ．SplitBoard[?] は，キーボードを分割し，従来のキーボードレイアウトよりも個々のキーの大きさを拡大することが可能になる．これらの手法は，ファットフィンガ問題を解決できるが，キーがスクリーンを専有しているためスクリーンオクルージョン問題が考えられる．本手法では，キーボードがスクリーンオクルージョンを低減するレイアウトを作成するため，上記の問題を解決可能である．

2.1.2 フリック操作を用いた文字入力手法

本節では，フリック操作を用いた文字入力手法における研究を示す．SwipeBoard[?] は，QWERTY キーボードを9つの領域に分割させ，フリック操作およびタップ操作によって領域を選択し，英文字を入力する手法である．具体的には，最初に9つの領域を，タップ操作および8つの方向に対するフリック操作にて選択する．次に，領域の中から入力したい文字をタップ操作およびフリック操作を行い文字を決定する．下岡ら [?] は，スマートウォッチを対象としたアイズフリー日本語文字入力手法は，指2本を用いてタップ操作および4方向のフリック操作を行うことによって文字入力を行う．さらに，スクリーン上にボタンは存在せず，操作

する指の本数，およびフリックする方向にて入力を識別する．そのため，アイズフリー入力
が実現可能である．

2.1.3 スワイプジェスチャを用いた文字入力手法

WatchWriter[?] は，QWERTY キーボード上にてスワイプジェスチャを行い，そのスワイプ
した軌跡からフレーズを推測する．そして，推測されたいくつかのフレーズから入力したい
フレーズ選択することによって文字を入力する手法である．ShuttleBoard[?] は，子音と母音の
15 個のキーにて構成されたキーボードレイアウト上をスワイプジェスチャを行うことによっ
て，かな文字入力を行う手法である．

これらの手法は，スクリーン上にて複雑な操作が必要とする．そのため，本研究ではスク
リーン上の操作は簡単なタップ操作のみしか行わないため，複雑な操作が必要ない．さらに，
物理的なインタフェースである竜頭を用いているため，より正確に入力が可能である [?].

2.2 スマートウォッチの竜頭を用いた研究

本節では，スマートウォッチの竜頭を用いた研究を示す．

2.2.1 スマートウォッチの竜頭を用いたインタラクション性能の調査

Frederic ら [?] は，スマートウォッチのタッチ入力，竜頭の回転操作における入力，および
ベゼルの回転入力の 3 つの異なる入力技術を対象とした 1 次元リスト選択のタスクの性能評
価を行った．結果として，ほとんどのユーザは竜頭操作を使いやすいと感じた．よって，ス
martウォッチのインタラクションにおける手段として，竜頭を用いることは有用である．

2.2.2 スマートウォッチの竜頭を用いた文字入力手法

CrownBoard[?] は，スマートウォッチの竜頭をもちいた英文字入力手法である．スマー
トウォッチのベゼルの周りの 8 つの領域にアルファベット郡を割り当てる．そして，スマー
トウォッチの竜頭を用いて，8 つの領域に割り当てられたアルファベット郡を選ぶ．選ばれた領
域のアルファベット郡から単語を推測し，ユーザは，推測された単語を選択することにより
入力を行う．CrownBoard は，竜頭を用いてタッチスクリーン上の複雑な操作を用いず単語を
入力が可能である．CrownBoard は，英文字における文字入力手法であり，竜頭を用いたかな
文字入力手法の研究はない．そのため，本研究は竜頭を用いたかな文字入力手法を実装し評
価を行う．

第3章 システム

本システムは竜頭の回転および簡単な画面操作を用いたかな文字入力手法である。本章では、かな文字入力システムにおける概要、および実装を述べる。

3.1 竜頭を用いた文字入力の方法

ユーザは、竜頭を上下に回転させることによって子音選択および母音選択を行う。また、スマートウォッチ上の画面をタップすることによって各選択操作を確定する。

3.1.1 子音選択

子音の選択には、竜頭を上下に回転させ操作する。初期状態は、「あ行」が選択されている。竜頭を上下に回転させることによって図 3.1 をもとに初期状態から遷移する。具体的には、竜頭を上回転させることによって時計回りに遷移し、竜頭を下回転させることによって反時計回りに遷移する。また、子音の選択状態が変化する毎に、スマートウォッチ上の振動センサを用いてユーザへ振動フィードバックを付与する。スマートウォッチ上の画面をタップすることによって、子音の選択を確定する。

3.1.2 母音選択

母音の選択には、竜頭を上下に回転させ操作する。子音選択によって選択された母音が画面に表示される（図??図を作成する）初期状態は、選択された子音に対応する「あ段」が選択されている。竜頭を上下に回転させることによって図 3.2 をもとに初期状態から遷移する。（図 3.2 を大きくする）具体的には、竜頭を上回転させることによって時計回りに遷移し、竜頭を下回転させることによって反時計回りに遷移する。また、母音の選択状態が変化する毎に、スマートウォッチ上の振動センサを用いてユーザへ振動フィードバックを付与する。スマートウォッチ上の画面をタップすることによって、母音の選択を確定する。

3.2 実装

3.2.1 システム環境

以下の環境を用いて実装を行っている。

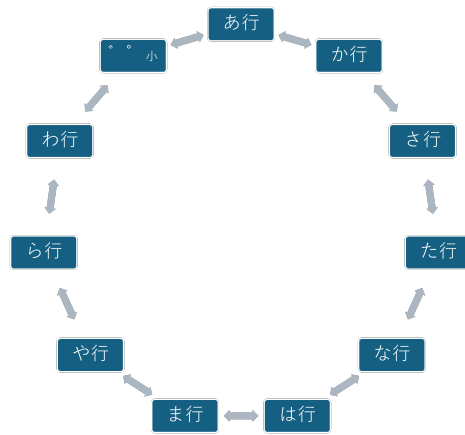


図 3.1: 子音選択の遷移図

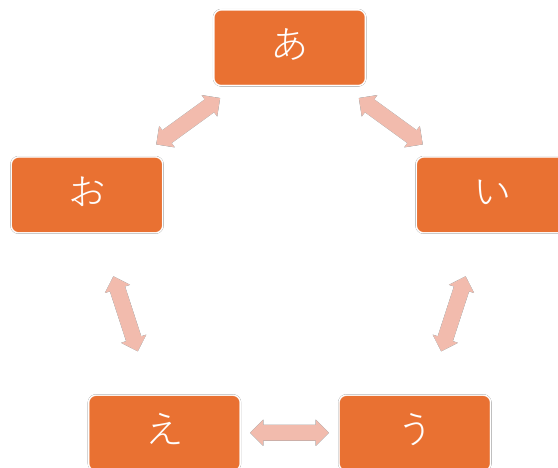


図 3.2: 母音選択の遷移図



図 3.3: 本システムのレイアウト. 左図：三文字分のスペースにて構成されているレイアウト（レイアウト 1），中央図：1 文字分のスペースにて構成されているレイアウト（レイアウト 2），右図：入力欄のみにて構成されているレイアウト（レイアウト 3）

- 装着型デバイス: Apple Watch
 - Model: Apple Watch Series 8
 - OS: watchOS 9.6.2
- データ受信機: iPhone
 - Model: iPhone 14
 - OS: iOS 16.6.1
- 実行環境
 - macOS: Ventura 13.51
 - プロセッサ: 3.5G Hz デュアルコア Intel Core i7
 - RAM: 16 GB 2133 MHz
 - GPU: Intel Iris Plus Graphics 650 1536 MB
 - エディタ: Xcode 14.3.1
 - ライブラリ: SwiftUI

3.2.2 レイアウト

システムを改良するため変更予定. なので図 3.3 も変更するので見なくて大丈夫です

スクリーンオクルージョン問題を解決するため図 3.3 に示すレイアウトを作成した. 本レイアウトには, 3 文字分のスペースにて構成されているレイアウト（以降, レイアウト 1）, 1 文字分のスペースにて構成されているレイアウト（以降, レイアウト 2）, および入力欄のみにて構成されているレイアウト（以降, レイアウト 3）がある. 本レイアウトは, 既存手法と異なりキーボードを必要としないため, スクリーンオクルージョンを改善する.

第4章 実験

本研究にて実装する文字入力手法の性能を評価するため、著者実験を行った。各レイアウトに対し、文字入力速度およびエラー率を計測し実験を行った。

4.1 評価指標

評価指標は、かな文字入力速度を測定するための指標である Character Per Minute(以下, CPM), および入力された文字のエラー率を測定するための指標である Character Error Rate(以降, CER)を使用した。

4.1.1 文字入力速度 (CPM)

文字入力速度である CPM を以下の 4.1 式に示す。それぞれの変数は入力および削除されたすべての文字 T の長さ $|T|$, および最初のキーの操作（子音選択）から、最後のキーの操作までに経過した時間 S である。

$$CPM = \frac{|T| - 1}{S} \times 60 \quad (4.1)$$

4.1.2 エラー率 (CER)

エラー率である CER は、入力および削除されたすべての文字 T および入力するべきの文字とのレーベンシュタイン距離を、入力するべき文字の文字数にて正規化する。

4.2 実験手順

実験の環境として、著者は利き手ではない左手にスマートウォッチを装着し、座りながら実験を行った。本手法の性能を評価するため、6 から 8 文字程度のフレーズを 60 個を用意した。**フレーズセットを付録に追加**。作成したフレーズセットから、ランダムに 30 個のフレーズを重複なく選び、入力した。1 つのフレーズを入力した後に、スマートウォッチのスクリーン左側に位置するボタンを押すことによって次のフレーズに移行する。すべての入力が完了した後に、1 分の休憩を行った。これを 1 セッションとして、各レイアウトに対して 2 セッションを

行った。また、本セッションを行う前に、練習セッションを行った。これは、各レイアウトに対して行った。練習セッションでは、本セッションとは違うフレーズを用いて練習を行った。

4.3 結果

システム改良後、著者実験をやり直し後記載（前回の結果：レイアウト 1 25CPM,0.044, レイアウト 2 23CPM,0.03, レイアウト 3 23CPM,0.07）

4.4 考察

システム改良後、著者実験をやり直し後記載

第5章 議論

本章では，本手法に関する改善案，今後の課題を述べる．本システムでは，キーの選択の際に振動フィードバックが生じるが，キーが遷移した際にラグが生じていた．さらに，キーの遷移がアニメーションのように動かないため，キーが遷移しているか分かりづらい．このような問題を解決するためにシステムを改良する予定である．システム改良後，著者実験をやり直し後記載

第6章 結論

著者実験をやり直したあと，記載

謝辭

著者論文リスト

参考文献

参考文献を載せる