

2022 年度

筑波大学情報学群情報科学類

卒業研究論文

題目

HMD 前面を用いた VR 用フリック文字入力手法

主専攻 知能情報メディア主専攻

著者 山口 泰生

指導教員 志築文太郎、川口一画、高橋伸

概要

Virtual Reality（以降，VR）における文字入力手法として，仮想空間に表示された仮想キーボードにおいてフリック入力を行う文字入力手法が研究されている．しかし，このフリック入力による VR 用の文字入力手法には普段スマートフォンにおいて行うフリック入力と比べ，入力速度および入力精度が低下するという問題がある．この問題の原因の一つとして，入力の際の触覚フィードバックが無いことがあげられる．そこで，本研究においては，ユーザが Head Mounted Display（以降，HMD）の前面に装着したタッチデバイスを触れることによって触覚フィードバックを得ることができる VR 用のフリック文字入力手法を示した．また，本手法の入力性能を向上させるための入力支援を付与した手法も併せて設計した．本手法，本手法に入力支援を付与した手法および従来の VR 用のフリック文字入力手法の文字入力性能の比較実験を行った結果，仮想キーボードのキーに対応するタッチデバイス上の領域の調整を行う入力支援を付与した手法が最も文字入力性能が優れていた．

目次

第1章 序論	1
1.1 背景	1
1.2 目的とアプローチ	3
1.3 貢献	4
1.4 本論文の構成	4
第2章 関連研究	5
2.1 VR 内のフリック入力に基づく文字入力手法	5
2.2 触覚フィードバックを付与した VR 内の文字入力手法	6
2.3 HMD に取り付けられた物理的な入力デバイスを用いた研究	6
2.4 固有感覚に基づく背面入力の研究	7
第3章 HMD の前面を用いたフリック文字入力手法	8
3.1 入力方法の検討	8
3.2 本手法の入力方法	8
3.3 入力の支援	9
3.3.1 支援1：指のホバー位置の表示	9
3.3.2 支援2：ユーザの固有感覚に基づく入力領域への調整	10
第4章 試作システム	11
4.1 実装	11
4.1.1 Android アプリケーションのシステム	11
4.1.2 Unity アプリケーションのシステム	11
4.1.3 キーボードの UI	11
第5章 予備実験	13
5.1 キーの配置の調整	13
5.1.1 キーの配置の条件	13
5.1.2 キーの入力タスク	13
5.1.3 入力タスクの結果	15
5.1.4 キーの配置	16
5.2 入力領域の調整	16
5.2.1 データの収集	16

5.2.2	モデルの作成および学習	17
5.2.3	モデルの評価	19
第 6 章	比較実験	22
6.1	比較手法	22
6.2	手順	22
6.3	実験結果	22
6.3.1	入力速度 (CPM)	23
6.3.2	エラー率 (TER)	23
6.4	考察	24
第 7 章	議論および今後の課題	25
7.1	複数人による実験	25
7.2	入力領域を調整する機械学習モデルの改良	25
7.3	学習に用いるデータの数	25
7.4	キーの大きさ	25
第 8 章	結論	27
	謝辞	28
	参考文献	29
	著者論文リスト	32
付 録 A	実験において用いた課題文	33

目 次

1.1	Meta Quest に搭載されている 50 音キーボード.	2
1.2	VR 内において用いられるフリックキーボード.	2
1.3	本手法における操作の外観.	3
3.1	本手法の入力時の様子. ユーザがタッチデバイスに触れると, VR 内に表示される仮想キーボード上に, 入力位置がポインタとして示される. また, ユーザは表示される仮想キーボードを裏からフリックするように入力する.	9
3.2	タッチ位置を表示するポインタを表示を与えて入力を行った際の 160 個の入力位置. 四角の黒線はキーの位置を示し, 各点を囲む楕円は 95 %の信頼楕円を示す.	10
4.1	仮想キーボード上に示されるポインタ. (a) 指の接触時の位置を示す赤色のポインタ. (b) 指のホバー位置を示す青色のポインタ.	12
5.1	条件 1: キー同士の横の間隔を 0.375 cm および縦の間隔を 0.375 cm としている.	14
5.2	条件 2: キー同士の横の間隔を 1 cm および縦の間隔を 1 cm としている.	14
5.3	条件 3: キー同士の横の間隔を 2 cm および縦の間隔を 1 cm としている.	14
5.4	条件 1 における 400 回の入力タスクによる入力位置.	15
5.5	条件 2 における 400 回の入力タスクによる入力位置.	15
5.6	条件 3 における 400 回の入力タスクによる入力位置.	16
5.7	仮想キーボード上に表示されるキーの位置と調整後の入力領域のイメージ.	17
5.8	モデル A におけるエポックごとの学習曲線の推移.	18
5.9	モデル A におけるエポックごとの精度 (Accuracy) の推移.	18
5.10	モデル B におけるエポックごとの学習曲線の推移.	18
5.11	モデル B におけるエポックごとの精度 (Accuracy) の推移	18
5.12	モデル A の混合行列.	19
5.13	モデル B の混合行列.	20
5.14	モデル B を用いて変換した入力領域. 各色は一つのキーの入力領域を示す. また四角の黒線はキーの表示位置を示す.	21
6.1	各手法の入力速度 (CPM) の平均を示した棒グラフ. エラーバーは標準偏差を示す.	23

6.2 各手法のエラー率 (TER) の平均を示した棒グラフ. エラーバーは標準偏差を示す.	23
--	----

表 目 次

A.1 実験において用いた課題文	33
----------------------------	----

第1章 序論

Virtual Reality (以降, VR) とは「本質的あるいは効果としては現実であるもの」[1]とされ, コンピュータによって生成された映像および音声によって, 現実に近い感覚をユーザに提示する. VR 内においてユーザは仮想的なオブジェクトを見ることができ, さらにコントローラ, もしくは自身の身体を用いてそれらとインタラクションすることができる.

1.1 背景

VR にはテキストチャット, メールおよび検索などの文字入力が必要とするコンテンツがある. ユーザは VR 内において文字入力を行う際, HMD により視界が遮られるため, 周りにある物理的な物理キーボード, あるいは他の文字入力デバイスを使用することが困難になる. そのため, VR 内における文字入力には図 1.1 のような仮想的な仮想キーボードが用いられる. ユーザは特定のアクションを行うことによってこの仮想キーボード上のキーを選択することにより文字入力を行う. この時, 仮想キーボードのキーの選択にはいくつかの手法がある. その中には VR ヘッドセットに付属するコントローラを用いる手法がある. この手法においては, ユーザは直接仮想キーボード上のキーに触れるもしくはコントローラから伸びるレイをキーへ向けることによって選択を行う [2].

しかし, 最近になると, LeapMotion [3] のようなユーザの手をハンドトラッキングして VR 内に投影するデバイスの登場によって, VR 内においてもユーザ自身の手腕によるインタラクションを行うことができるようになった. また, それに従い, VR 内において必ずしもコントローラが必要になる場面はなくなったため, 文字入力のためだけにコントローラを把持することは好まれない. そのため, コントローラを使わずに, 手の動きをトラッキングして文字入力を行う手法が提案されている. しかし, VR は日常生活において常に使うものではない. したがって, ユーザが VR 内における文字入力専用の文字入力手法を習得することは慣れるまでに時間がかかる, もしくは労力が必要となる. そのため, VR 内における文字入力は現実の文字入力において扱うような QWERTY 入力およびフリック入力を模したものである方が習得コストが低い.

現実における文字入力として QWERTY 入力は一般的に行われているが, 長尾ら [4] の調査によると近年の日本の若年層においては, 日本語の文章を入力するときに QWERTY 入力よりも, フリック入力の方が好まれる. また, 日本語入力において, 一字入力する際のキー入力の回数は QWERTY キーボードによるローマ字入力よりもフリック入力の方が少なくなる場合が多い. そこで, 本研究においては, フリック入力に基づいた VR 内における日本語か



図 1.1: Meta Quest に搭載されている 50 音キーボード.

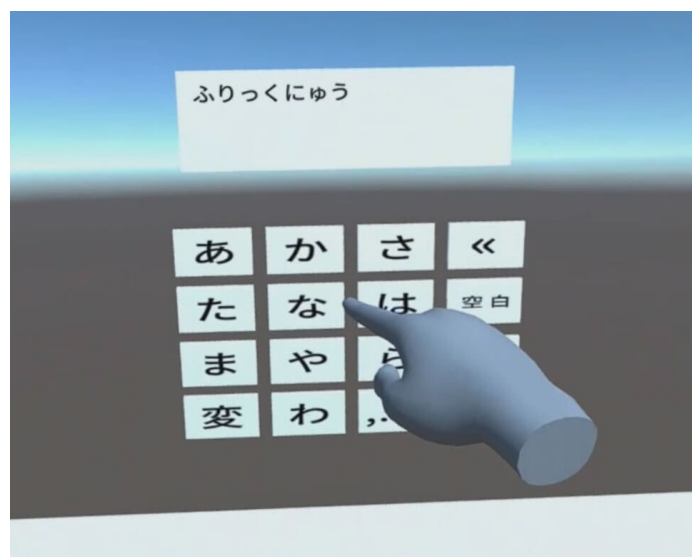


図 1.2: VR 内において用いられるフリックキーボード.



図 1.3: 本手法における操作の外観.

な文字入力手法を示す.

VR 内におけるフリック文字入力とは VR 内に図 1.2 のような仮想キーボードを表示し、それを手指を用いて文字入力を行う手法 [5, 6, 7] が提案されている. しかし、仮想キーボードを用いた文字入力は現実世界のスマートフォンの画面を用いたフリック文字入力と比べて入力速度および入力精度が低下するという問題がある. この問題の理由の一つとしてユーザが仮想キーボードを用いて文字入力する際に、触覚のフィードバックを得られないことがあげられる. さらに、フリック入力において、ユーザはキーボードに触れた指をスライドさせる操作を行うため、ユーザに対してキーボードに指が接触しているという触覚フィードバックを与えることは重要である.

1.2 目的とアプローチ

本研究において、我々は VR 内におけるフリック文字入力の入力速度および入力精度の向上を目的として、図 1.3 のように HMD の前面に取り付けたタッチデバイスを用いてフリック文字入力を行う文字入力手法を示す. 本手法において、ユーザは HMD の前面に取り付けたタッチデバイスをフリックキーボードの面として用いることにより、ユーザは外部デバイスを把持および装着を行わずにフリック入力の際に触覚フィードバックを得ることができる. また、キー入力を行いやすくするために仮想キーボード上にタッチデバイスに触れる前の指のホバー位置を示す入力支援およびタッチデバイス上のキーの入力に対応する領域（以降、入力領域）を事前に収集したユーザの固有感覚 [8] に沿った入力位置のデータに基づいて調整を行う入力支援をそれぞれ設計する.

1.3 貢献

本研究の貢献は以下の通りである．

- HMD の前面に取り付けたタッチデバイスを用いたフリック入力手法を示した．
- HMD の前面に取り付けたタッチデバイスを用いたフリック入力を行う際の適切なキー配置を示した．
- 比較実験において，HMD の前面に取り付けたタッチデバイスを用いたフリック入力の方が従来の手法より入力速度および入力精度が高いことを示した．

1.4 本論文の構成

本論文の構成は以下の通りである．第 1 章においては，本研究の背景，目的とアプローチおよび貢献を示した．第 2 章においては，本研究に関連する研究を述べ，本研究の位置付けを示す．第 3 章においては，本研究において示す手法および入力支援の方法を述べる．第 4 章においては，本手法の試作システムの実装を述べる．第 5 章においては，入力位置に基づく入力に適した VR 内におけるキーの配置およびタッチデバイス上におけるキーに対応する入力に適した領域を調査および結果を述べる．第 6 章においては，本手法および入力支援による文字入力性能を調査するために実施した実験を述べる．第 7 章においては，評価実験の結果に基づいて本手法の問題点および改善案について議論し，今後の課題を述べる．第 8 章においては，本研究の結論を述べる．

第2章 関連研究

本章においては、本研究の関連研究として VR 内におけるフリック入力手法に基づく文字入力手法、触覚フィードバックを付与した VR 内における文字入力手法、HMD に取り付けた物理デバイスを用いた研究および固有感覚に基づく背面入力の研究を示す。また、それらに対する本手法の位置づけを述べる。

2.1 VR 内のフリック入力に基づく文字入力手法

本節において、VR 内におけるフリック入力に基づく文字入力手法に関する研究を述べる。喜多ら [5] は仮想の QWERTY キーボードおよび仮想のフリックキーボードに対して文字入力タスクを行うことによって、入力速度、正確性および操作感を比較した。その結果、仮想のフリックキーボードによる入力は仮想の QWERTY キーボードによる入力に対して入力速度および正誤率において劣ったと報告した。しかし、この手法において、いずれの手法においても入力の際のフィードバックを実装しておらず、本来の性能を示せていない可能性があるとしている。福仲ら [6] はキーに触れた際の母音キーの展開、キーの色の变化およびキーの押し込み表現という視覚フィードバックを付与し、その効果を調査した。調査の結果、キーの色の变化の視覚フィードバックが入力速度に及ぼす効果はこの実験からは不明だった。また、キーを押し込んでから指を引き上げる際に指先が別の領域に移動することによる誤入力が多かったと報告している。

これらの手法はフリック入力の際に触覚フィードバックがない。対して、本手法においては HMD の前面に取り付けたタッチデバイスに触れることによってフリック入力の際に触覚フィードバックを付与する。

川口ら [9] は VR 内において片方の手の掌に表示したフリックキーボードを他方の手の人差し指によって操作する手法を示した。この手法において、ユーザは自らの手に触れることにより、触覚フィードバックを得ることができる。ただし、この文字入力手法は手形状の誤認識の多発により、入力速度を示す Characters Per Minute (CPM) および入力精度を示す Corrected Error Rate [10] が視覚フィードバックのみの VR フリック入力手法と比較して低かった。

一方、本手法は HMD の前面に取り付けたタッチデバイス上のタッチ位置を VR 空間の仮想キーボード上に反映することにより、正確な入力を実現する。

小澤ら [7] は、仮想のフリックキーボードの子音キーをつまむ動作によって子音を選択、また、つまむ動作のまま指を移動させることによって文字を選び、指を離すことで文字を入力する手法を示した。

この手法はつまむことにより、仮想のフリックキーボードを用いて触覚フィードバックの伴う文字入力を行っているが、つまむ動作による入力は現実におけるフリック入力の操作方法と異なる。

2.2 触覚フィードバックを付与した VR 内の文字入力手法

本手法においては文字入力の際の触覚フィードバックの付与のために HMD の前面に取り付けたタッチデバイスを用いた。本節において、VR 内における文字入力に対して触覚フィードバックを付与した文字入力手法に関する研究を述べる。Henderson ら [11] は、座位状態において大腿部に取り付けた、もしくはポケットの中に入れたスマートフォンを VR 内の文字入力デバイスとして用いる手法を示した。この手法において、ユーザはスマートフォンをタッチおよびスワイプして VR 内の仮想キーボード上に表示されるカーソルを操作することによって文字入力を行う。Hirai ら [12] はユーザの周囲の物体の表面を入力面として利用することによって、VR 内における触覚フィードバックを伴う QWERTY キーボード文字入力手法を示した。この手法においては、HMD に搭載されているトラッキング機能を用いて手の位置と姿勢および物体の表面を認識している。Kim ら [13] は把持したスマートフォンを用いて VR 内における文字入力を行う文字入力手法を示した。この手法はスマートフォンのホバリング機能によって取得した親指のホバー位置を VR 内に表示される仮想キーボード上に示すことによって正確に文字入力を行う。この手法において入力速度は従来手法と有意な差がなかったが、ホバリング機能を用いた入力はユーザの指が不意に画面に触れるおよびホバー感知域外に出ないように注意する必要がある、ユーザにストレスがかかると報告している。Wu ら [14] は VR 内の仮想キーボードのキーを押すと同時に、指先にマイクロスピーカを取り付けた手袋を振動させることによって文字入力に触覚フィードバックを付与した。この手法においては実際のキー押下時の触覚を再現することにより、VR 内におけるキーボード入力の没入感を高め、入力性能が向上したと報告している。

これらの研究においては触覚フィードバックを付与する際に周りの環境に依存する、もしくは外部デバイスの把持および身体への装着が必要である。これに対して、本手法においては環境によらず、また、外部デバイスの把持および身体への装着の必要がない。

2.3 HMD に取り付けた物理的な入力デバイスを用いた研究

本研究においては、HMD の前面に取り付けたタッチデバイスによるフリック入力手法を示した。本節において、HMD に取り付けた物理的な入力デバイスを用いた研究を述べる。Gugenheimer ら [15] および Lee ら [16] は HMD の前面に取り付けたタッチデバイスを用いた入力手法を示した。彼らはこの入力手法において、HMD の前面を用いた入力手法、VR 内に表示されるタッチデバイスの表示位置および入力の特性を調査した。また、Gugenheimer ら [15] は HMD の前面に取り付けたタッチデバイスを用いた際の入力方法として、LandOn、LiftOff、および PressOn を提案し、調査を行った。LandOn において、ユーザはスマートフォ

ンにおけるタッチ入力と同様に、タッチした位置のターゲットを選択する。ただし、ユーザはHMDを装着しているため、指がタッチデバイスに触れる正確な位置を把握できない。そのため、この入力手法は細かいターゲットの選択には向かない。細かいターゲットの選択の場合、タッチにより現在の位置を示し、ターゲットまで指が移動したのちに、特定のアクションを行うことによってターゲットの選択を行う手法が LiftOff および PressOn として示されている。LiftOff において、ユーザはタッチした位置ではなく、離れた位置によってターゲットを選択する。また、PressOn においては画面の押し込みにより選択が行われる。LiftOff および PressOn は LandOn に比べて入力速度が低下する。Hutama ら [17] は HMD の側面部に取り付けた物理的な QWERTY キーボードを用いた文字入力手法を示した。この手法において、従来のコントローラを用いたポインティングによる文字入力と比べ文字入力速度が向上した。しかし、この手法はユーザがキーボードを見ることができないためホームポジションの把握が難しいことが報告されている。Tseng ら [18] は HMD の前面に入力を行えるウィジェットを設置し、それを用いた入力システムを示している。この研究において、ユーザが HMD の前面において操作を行う際、ウィジェットを取り付けた台座を前方に延ばすことにより、入力時の目の疲労を抑えることができたと報告している。

これらの研究において、HMD の前面を用いた入力、また、HMD に取り付けられた QWERTY キーボードを用いた入力の調査が行われたが、HMD の前面に取り付けたタッチデバイスを用いたフリック入力文字入力の調査は行われていない。

2.4 固有感覚に基づく背面入力の研究

人間には、固有感覚 [8] という、自身の身体の動作による手足の相対的な位置（位置覚）および運動の方向（運動覚）がわかる運動感覚が備わっている。身近な例として、人間は目をつぶった状態において、自分の鼻に触れることができる [19]。この時の身体部位の位置や動きを認識するための感覚が固有感覚である。本手法において、ユーザが HMD の前面に対して行う入力はユーザの固有感覚に従う [15]。そのため、固有感覚に基づくデバイスの背面における入力に関する研究を述べる。Wolf ら [20] は板状のデバイスを両手で把持し、その背面に触れた際の表面への視覚的なフィードバックが、背面の入力に対してどのような影響を与えるかを調査した。調査の結果、参加者は、指の位置の視覚的なフィードバックがない場合においても、ある場合と同様に正確にターゲットを選択することができた。また、視覚的フィードバックを取り除いても、作業負荷の著しい増大は確認されなかったと報告している。Coskun ら [21] は背面にタッチセンサを搭載するタブレット端末を両手で持った状態における、背面に位置する指を含めた 10 本の指によるテキスト入力手法を示した。ユーザ実験の結果、タブレット端末によって指が隠れているにもかかわらず、8.6 WPM およびエラー率が 17.52 % において文字入力を行うことができたと報告している。

これらの研究においては、タブレットおよびスマートフォンを把持した状態における背面入力を行っているが、本手法は HMD の前面において同様の操作を行う点においてこれらの研究と異なる

第3章 HMDの前面を用いたフリック文字入力手法

本章において、HMDの前面に取り付けたタッチデバイスを用いるVR用フリック文字入力の概要、入力方法および入力の支援を述べる。

3.1 入力方法の検討

HMDの前面を用いてフリックする際の入力方法を検討する。フリック入力において、ユーザはまず入力したい文字の「あ」段のキー（以降、子音キー）に触れる。次に、指をタッチデバイスに触れたまま入力したい文字の方向に指をフリックさせる。最後に指を離すことによって文字入力を完了する。この操作手順を本手法において行うために、Gugenheimer [15] が調査した LandOn, LiftOff および PressOn の入力方法を参考にする。LandOn の場合、ユーザはタッチしてフリックするという普段のフリックと同様に一回のタッチによって文字入力を行うことができる。しかし、LiftOff において、ユーザは子音キーの選択とそれに続く母音の選択を行う際に、一回のタッチにおいて子音と母音の二段階選択を行う [22] もしくは子音キーの選択とそれに続く母音の選択のために二回のタッチを行うなどの普段行うフリック入力と異なる入力を行う必要がある。また、PressOn において、押し込む際に画面の揺れを引き起こすという問題に加え、ユーザは画面に常に触れることになるため、フリック入力の操作方法と異なる。よって、HMDの前面におけるフリック入力の方法として子音キーの選択に LandOn を用い、それに続く母音の選択に LiftOff を用いる。

3.2 本手法の入力方法

本手法において、ユーザはHMDの前面に取り付けたタッチデバイスの表面をフリック入力して文字入力を行う（図3.1）。タッチデバイスの領域と仮想キーボードは対応付けられており、文字入力時の指の位置はVR内の仮想キーボードに反映され、ユーザに示される。そのため、ユーザはフリック入力に合わせて、文字入力の様子を確認しながらキーと指の接触を感じさせる触覚フィードバックを得ることができる。この時、通常のフリック入力において、ユーザはキーボードの表側から指を奥行き方向に向けて入力するのに対し、本手法においては、ユーザは仮想キーボードの裏側から指を当ててフリック入力を行う。



図 3.1: 本手法の入力時の様子. ユーザがタッチデバイスに触れると, VR 内に表示される仮想キーボード上に, 入力位置がポインタとして示される. また, ユーザは表示される仮想キーボードを裏からフリックするように入力する.

3.3 入力の支援

ユーザは HMD の装着中において, 空中における指の位置を視認することができないため, 正確な位置を把握しきれない. また, ハンドトラッキング機能を搭載する HMD, もしくは LeapMotion [3] は HMD の前方における手指の位置を取得できるが, 本手法において行う HMD の前面へのタッチはトラッキングセンサとの距離が近すぎるによりセンサの死角に入るため, 本手法はトラッキングセンサを用いて空中における手指の位置を取得することができない. そこで, HMD の前面を用いたフリック入力を容易にするための入力の支援を考える.

3.3.1 支援 1: 指のホバー位置の表示

ホバー位置の表示による VR 用の文字入力の支援は Kim ら [13] によって調査された手法を用いる. この支援方法は指のホバー位置を取得できるタッチデバイスを用いることによって, ホバー位置を VR 内の仮想キーボード上に表示する. 実際に筆者がこの支援方法を用いたところ, 指のホバー位置の取得に遅延があり, 指のホバー位置を示すポインタが現実の指の移動より少し遅れてついてくるように感じた. また, タッチデバイスが指のホバーを認識できる奥行き方向の範囲は画面から 2 cm 程度であり, ユーザは指のホバー位置を把握するために常にその領域内に指を留めておく必要がある.

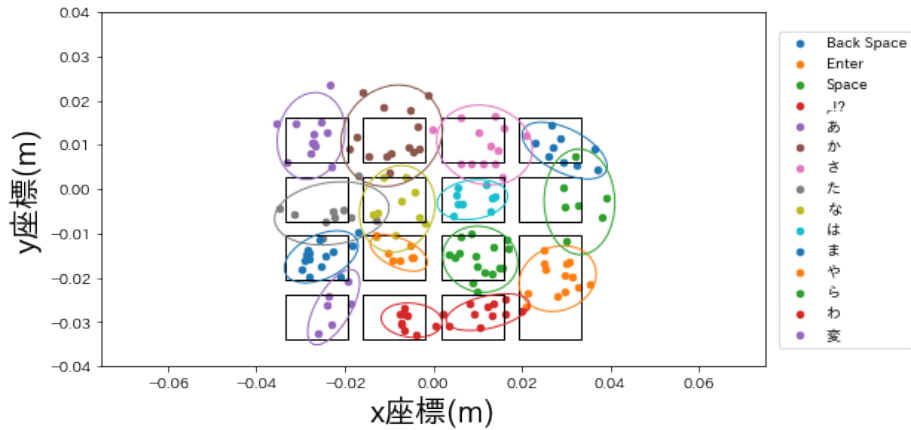


図 3.2: タッチ位置を表示するポインタを表示を与えて入力を行った際の 160 個の入力位置. 四角の黒線はキーの位置を示し, 各点を囲む楕円は 95 %の信頼楕円を示す.

3.3.2 支援 2: ユーザの固有感覚に基づく入力領域への調整

本手法において, ユーザは VR 内の仮想キーボードを見ながら, HMD の前面に取り付けられたタッチデバイスに対して固有感覚に従った入力を行う. ここで, ユーザは VR 内に表示されるキーの配置をもとにタッチデバイス上のキーに対応する場所にタッチを行うため, この時の入力位置の分布は VR 内に表示されるキーの配置に従う. 実際に著者がホバーのポインタの表示がない状態においてキーを入力した際の入力位置を図 3.2 に示す. 図 3.2 は四角の黒線をキーに対応する入力領域, 各色の点を入力位置, その点群を囲む楕円を 95 %の信頼楕円として示す. 95 %の信頼楕円とはその領域内に点群の真の平均が 95 %の確立において存在することを示す. この信頼楕円をもとにキーに対応するタッチデバイス上の領域 (以降, 入力領域) を考察する. 四角の黒線の領域に点が収まっていない箇所があることから, 固有感覚に基づいた入力位置と実際の入力領域にはズレがあることがわかる. そのため, 図 3.2 において四角の黒線として表されるタッチデバイスにおける入力領域を固有感覚に沿うように調整することによって入力がしやすくなると考えられる. また, 信頼楕円が重なる箇所はそれぞれのキーの入力位置の分布が明確には分かれていないことを示し, このような信頼楕円の重なりが生まれるようなキーの配置においてはユーザの入力したキーを正確に決定できない, もしくはユーザに誤入力を引き起こさせやすい. そのため, VR 内に表示する仮想キーボードのキーの配置についても, 信頼楕円が重ならないようなキーの間隔を調査する必要がある.

第4章 試作システム

本章においては，本手法の試作システムに用いたデバイスおよび試作システムを構成する Android アプリケーションと Unity アプリケーションの実装を述べる．

4.1 実装

タッチデバイスとして，指のホバーを取得可能なスマートフォンである Samsung Galaxy note3 Android 5.0 を用いた．また HMD に Oculus Quest を用いた．今回のシステムにおいてはスマートフォン上において動作する Android アプリケーションと PC 上において動作する Unity アプリケーションの間において通信を行う．HMD は PC 上の Unity アプリケーションにおいて処理される映像を映し出すために用いた．この時，通信には WebSocket を用いた．

4.1.1 Android アプリケーションのシステム

スマートフォンにおける入力処理するアプリケーションを Android Studio を用いて作成した．このアプリケーションは指の接触およびホバーの状態を判別し，それぞれの状態において，指の座標を取得する．また，取得した情報を Unity アプリケーションにリアルタイムにおいて送信する．

4.1.2 Unity アプリケーションのシステム

文字入力アプリケーションを Unity を用いて作成した．Unity 上には入力の様子を表す仮想キーボードが表示される．実装した仮想のフリックキーボードには一般的なフリックキーボードに備わっているキー入力の際にそのキーに付随して表示されるキーの展開表現を模倣した視覚フィードバックを付け加えた．また，ユーザがタッチした位置を図 4.1 に示すようなポインタによってキーボード上に表示する．この時，ポインタの色をタッチデバイスに触れている際に赤色とし，3.3.1 項の支援を用いる際はホバー時に青色となるようにした．

4.1.3 キーボードの UI

本手法のキーボードの UI を述べる．VR 内に表示される仮想キーボードを頭と HMD 前面の位置関係を保つように常に顔の正面に配置した．この時，仮想キーボードの大きさを

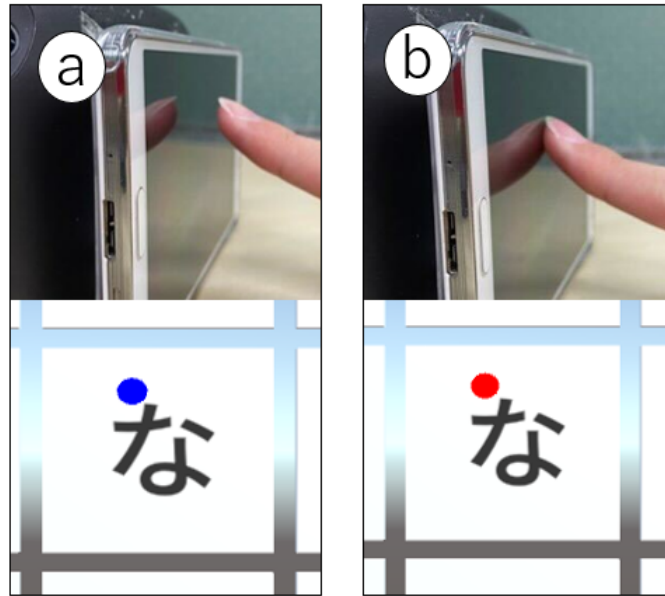


図 4.1: 仮想キーボード上に示されるポインタ. (a) 指の接触時の位置を示す赤色のポインタ. (b) 指のホバー位置を示す青色のポインタ.

Gugenheimer ら [15] の手法を参考にして、現実のタッチデバイスの大きさと同じ縦 8 cm、横 15 cm、また頭を中心から仮想キーボードまでの距離を 30 cm とした。また、入力中のテキストの位置をキーボードの直後に表示することにより焦点の遠近の移動を抑制することによってユーザの負担を減らす。キーは通常時に半透明とし、指がキーに触れた際に、接触中のキーを除いた他のキーを透明化する視覚フィードバックを付与する。

第5章 予備実験

本章において、3.3.2において述べた、固有感覚に沿った入力を行った際における入力領域の重なりが起こらないキーの配置および入力領域の調整を調査する予備実験を述べる。これらの予備実験の参加者は著者1人である。参加者は22歳の大学生の男性1名、右利きであり、フリック入力を日常において使用している。またVRの経験は日常生活において使用せず、研究活動を行う際に使用する程度である。

5.1 キーの配置の調整

本節において、調査するキーの配置の条件、実験に用いるキーの入力タスク、各条件における入力タスクの結果および本手法に用いるキーの配置を述べる。

5.1.1 キーの配置の条件

キーの配置の条件として、横の間隔および縦の間隔を変えた3種類を設けた。条件1は横の間隔を0.375 cm および縦の間隔を0.375 cm (図 5.1)、条件2は横の間隔を1 cm および縦の間隔を1 cm (図 5.2)、条件3は横の間隔を2 cm および縦の間隔を1 cm (図 5.3)とする。また、確定(Enter)キー以外のすべてのキーのサイズは横を2 cm および縦を1 cm とし、確定キーのサイズは条件1において横を2 cm および縦を2.375 cm、また、条件2および3において横を2 cm および縦を3 cm とする。なお、条件3はキーに触れた際に展開されるキーが、入力できる領域内における収まる状態において、キー同士が最も離れた配置である。

5.1.2 キーの入力タスク

各条件において、ランダムに各キーを入力するタスクを行い、それぞれのキーにおける入力位置を調べる。入力タスクの手順を述べる。まず、子音キー、特殊記号キー、文字変換キー、消去キー、スペースキーおよび確定キーの中から一つ入力する文字をランダムに選択し、参加者に入力するようVR内に表示する。次に参加者は指示された文字を入力する。この時、参加者には入力の際にタッチした位置を示すポインタのフィードバックが示されるが、指のホバー位置を示すポインタは示さない。これは、ホバー位置を示すポインタを表示すると、低速、かつ正確にキーの位置を入力することが可能になり、固有感覚に基づく入力位置を測定できなくなるからである。最後に、参加者が指を離し入力を完了させると次の入力すべきキー

の提示までに 1 秒間のインターバルを設ける．この一連の作業を一回の入力タスクあたり 50 回繰り返す．また，文字が指示されてから，参加者が入力するまでの時間を計測した．

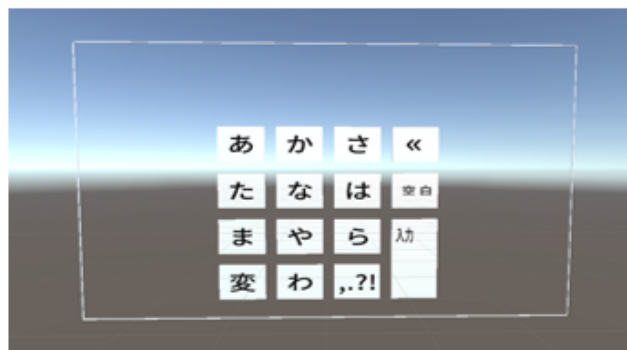


図 5.1: 条件 1：キー同士の横の間隔を 0.375 cm および縦の間隔を 0.375 cm としている．

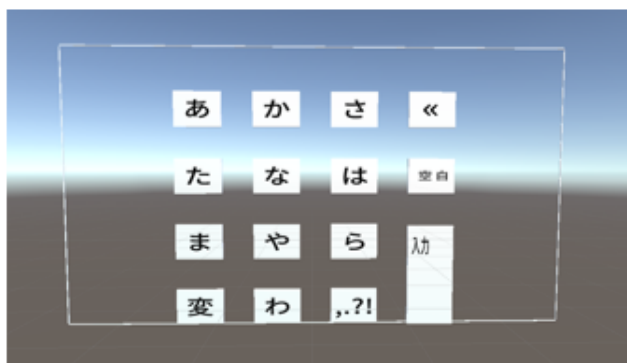


図 5.2: 条件 2：キー同士の横の間隔を 1 cm および縦の間隔を 1 cm としている．

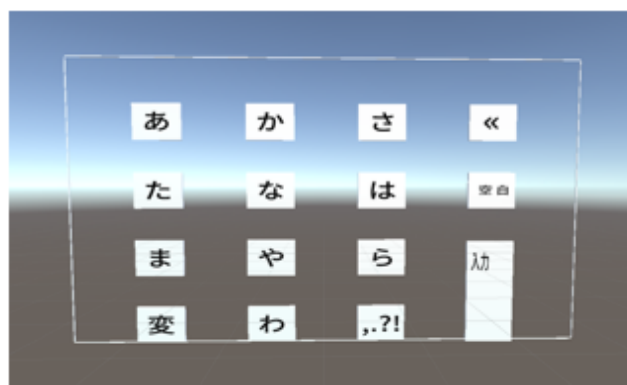


図 5.3: 条件 3：キー同士の横の間隔を 2 cm および縦の間隔を 1 cm としている．

5.1.3 入力タスクの結果

上述した3条件においてそれぞれ入力タスクを8回行った。1条件当たり合計400回の入力の結果を図5.4～5.6にそれぞれ条件1～条件3における入力位置の散布図として示す。ここで、図5.4は多くのキー同士において、また図5.6は「た」と「ま」および「は」と「ら」のキーにおいて信頼楕円の重なりが見られたが、条件2においては信頼楕円の重なる部分なかった。また、それぞれの条件における各信頼楕円の面積の総和は、条件1が 29.27cm^2 、条件2が 25.12cm^2 、条件3が 41.21cm^2 となった。

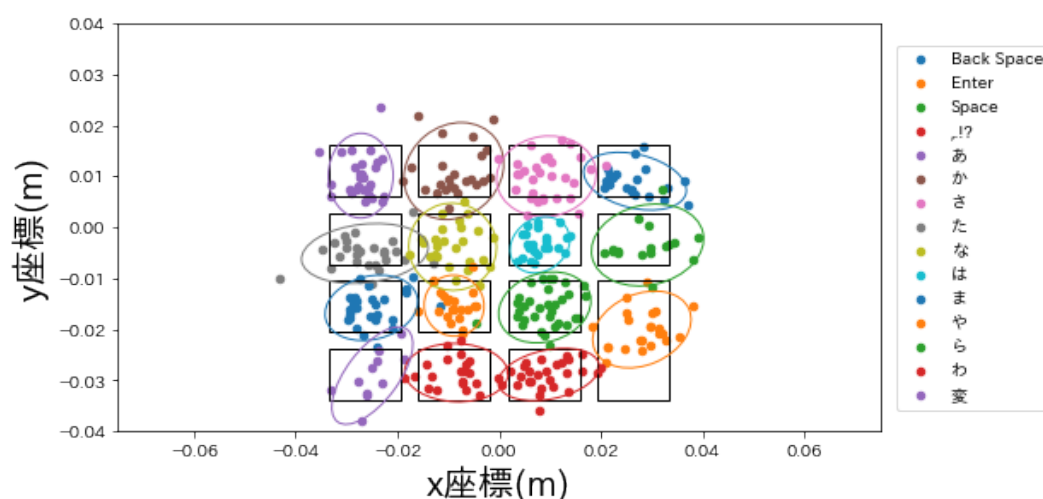


図 5.4: 条件 1 における 400 回の入力タスクによる入力位置.

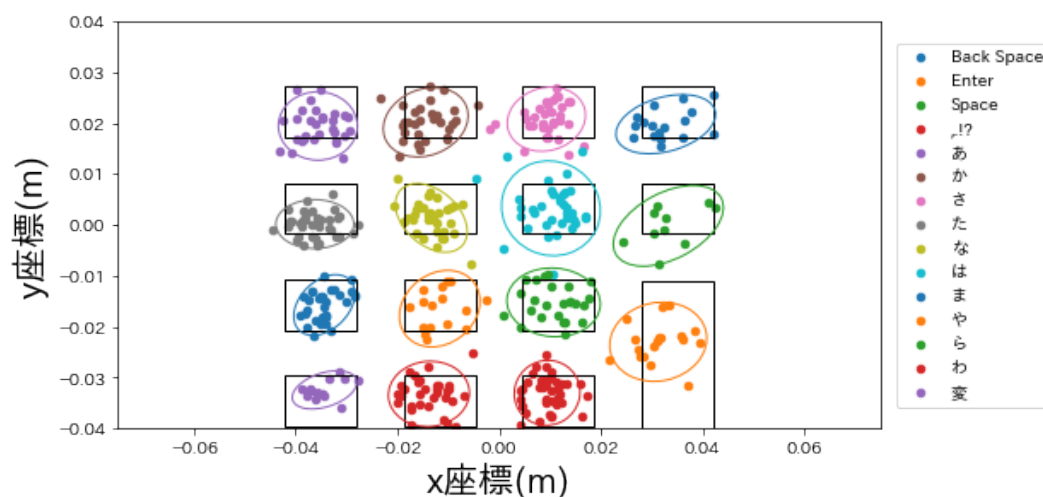


図 5.5: 条件 2 における 400 回の入力タスクによる入力位置.

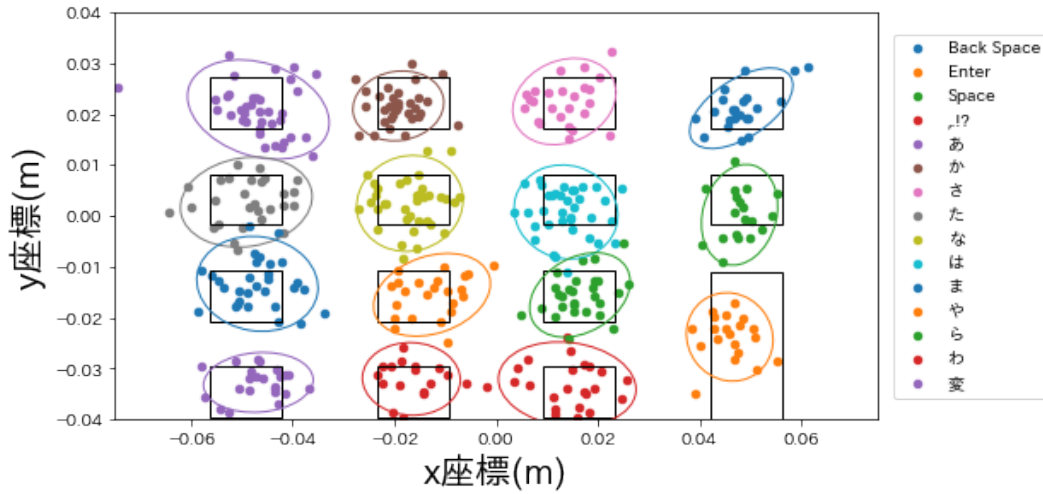


図 5.6: 条件 3 における 400 回の入力タスクによる入力位置.

5.1.4 キーの配置

これらの結果をもとに本手法における最適なキーの配置を考察する．条件 1 はキーの間隔が狭いことにより，隣のキーの信頼楕円と重なると考えられる．また条件 3 は信頼楕円の面積が他の条件と比べて大きい，これはキーとキーの間隔が大きくなることにより指の移動が大きくなったため入力時の位置がズレやすくなると考えられる．そのため，キーの間隔は信頼楕円が重なるほど小さくなく，また指の移動量が入力位置のズレを引き起こすほど大きくない条件 2 が適切であると考えられる．よって，本手法のキーの配置には，信頼楕円の重なりが最も少ない条件 2 を用いる．

5.2 入力領域の調整

本節においては，入力領域の調整に必要なユーザの入力位置のデータの収集，入力領域の調整を行う機械学習モデルの作成と学習，およびモデルの評価を述べる．

5.2.1 データの収集

各キーごとの入力領域を調整するために，提示したキーに対して，ユーザが実際に入力した位置とその真値であるキーの中心の位置を合わせたデータを収集する．また，今回の本手法に用いるタッチデバイスは指のホバー位置を取得できるため，ホバー位置も合わせて収集する．これにより入力する直前の指の軌道の情報についても把握することができる．ただし，取得できるホバー位置はタッチデバイス上の x および y 座標のみであり，タッチデバイスからの高さを取得することはできない．そのため，収集するデータはタッチデバイスに触れた

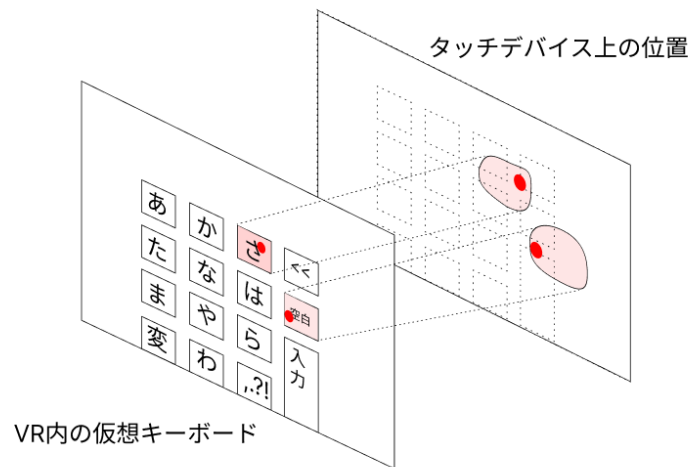


図 5.7: 仮想キーボード上に表示されるキーの位置と調整後の入力領域のイメージ。

実際の座標およびタッチデバイスに触れる直前の Unity のフレームレートの 25 フレームの内、5 フレームごとのそれぞれの x および y 座標の計 12 個の値とした。ここで、25 フレーム以内としたのは、入力位置から遠い位置の情報がノイズになると考え、また 5 フレームごととしたのは 4 フレーム間を開けることによって同様の値の連続を避けるためである。収集には 5.1.2 の入力タスクにおいて用いたアプリケーションをそのまま用いた。この時、入力タスク一回当たりの入力の回数を 50 回から 75 回に変更し、入力タスクを 20 回繰り返すことによって 1500 個の入力のデータを収集した。なお、データの偏りを避けるために入力タスクを行う間を 1 時間ほど開け、3 日間に分けて収集した。

5.2.2 モデルの作成および学習

入力位置から入力領域の調整する機械学習モデルを作成する。作成するモデルにおいては入力位置から予測を行うことにより、出力結果を 15 個のキーの内の一つとするクラス分類を行う。機械学習のモデルの作成および学習を Python において行うが、データの取得および予測結果の利用は Unity アプリケーションにおいて行う。そこで、実装の複雑さを解消するために作成したモデルを Unity アプリケーションに組み込み、予測を行うようにする。PyTorch は作成したモデルを ONNX 形式に変換および出力することができ、変換した ONNX 形式のモデルを Unity にインポートすることにより、Unity アプリケーションにおいてモデルを実行させることができる。そのため、PyTorch によって入力領域の調整を行う深層学習モデルを作成する。今回は 12 値を入力とする分類モデル（以降、モデル A）に加え、タッチした際の x および y 座標の 2 値のみを入力とする分類モデル（以降、モデル B）をそれぞれ作成し、それぞれのモデルを用いた際の入力性能を調査する。モデル A およびモデル B を作成する際に、それぞれのモデルにおいて、いくつかの深層学習モデルを作成した。各モデルのすべての候

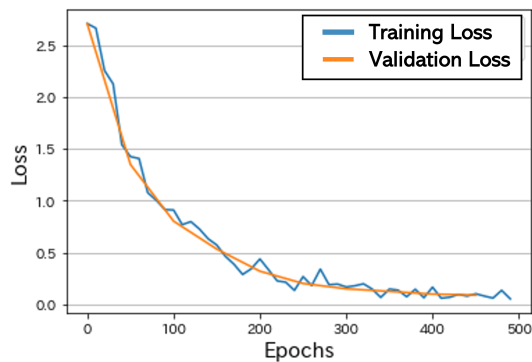


図 5.8: モデル A におけるエポックごとの学習曲線の推移.

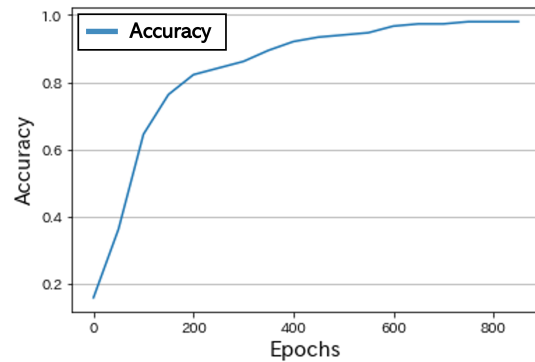


図 5.9: モデル A におけるエポックごとの精度 (Accuracy) の推移.

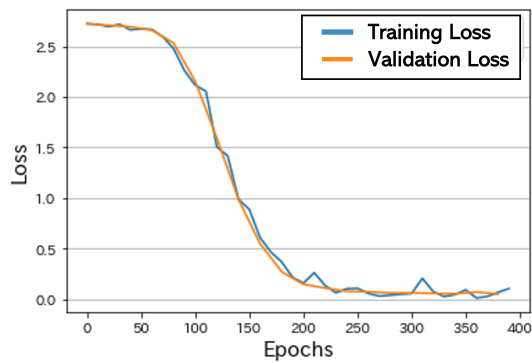


図 5.10: モデル B におけるエポックごとの学習曲線の推移.

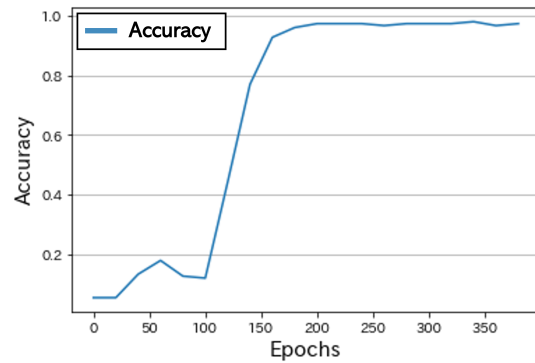


図 5.11: モデル B におけるエポックごとの精度 (Accuracy) の推移

補は活性化関数に ReLu 関数を適用させた全結合層（各層のパラメータの数は 8～64）を 3～5 個重ね、最後に log softmax 層をつけたものとした。これらの候補に対して、対応するデータを与えて学習させ、評価したところ、いずれも精度は 98 %程度を示した。そこで、候補として作成したモデルの中から最もパラメータが少ないモデルをそれぞれ選択し、モデル A を全結合層（各層のパラメータの個数は 15～16）を持つモデル、またモデル B を 3 層の全結合層（各層のパラメータの個数は 15～32）を持つモデルとした。なお、モデルの学習には収集した 1500 個のデータの内、その 8 割に相当する 1200 個を用い、残りの 300 個のデータを用いて評価および検証を行った。この時のモデル A の学習および評価を図 5.8 および図 5.9 に示し、モデル B の学習および評価を図 5.10 および図 5.11 に示す。

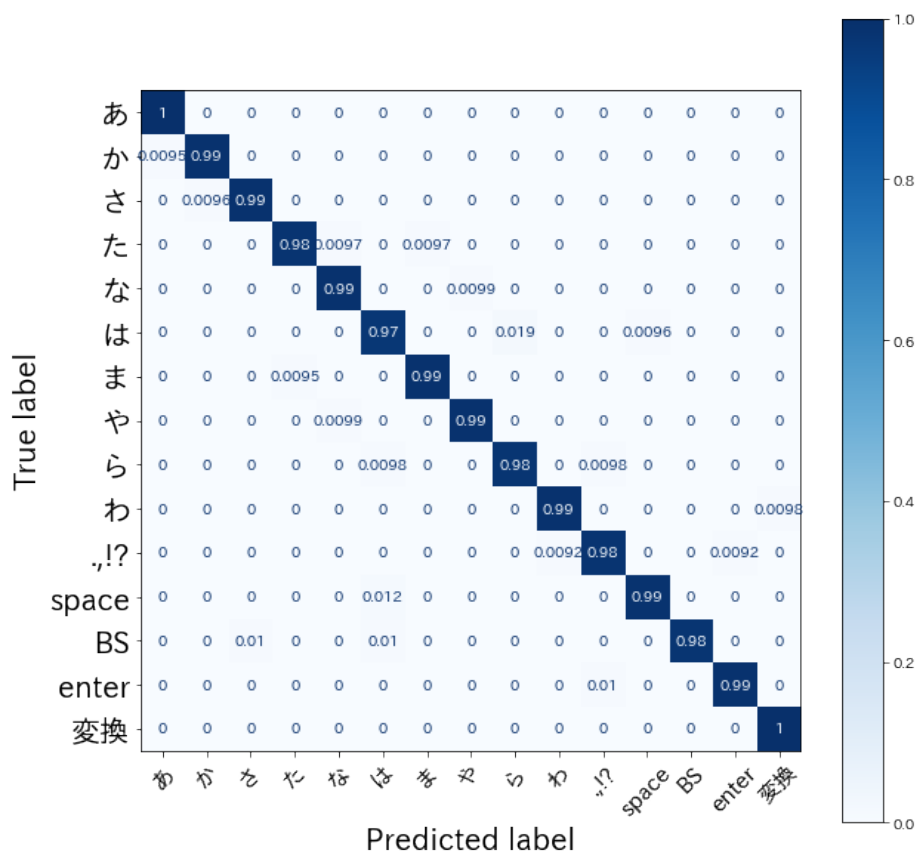


図 5.12: モデル A の混同行列.

5.2.3 モデルの評価

作成した二つのモデルの評価を行う。モデル A において精度は 0.989, 再現率は 0.987 および F1 スコアは 0.987 であった。また, モデル A における各キーの入力の混同行列を図 5.12 および学習時における学習曲線を図 5.8 に示す。図 5.8 において青色の実線は損失関数を示し, また, 黄色の実線は評価関数の値を示している。この時, 評価関数の示す値が学習が進んでも損失関数と乖離しないことから, モデルが学習データに過剰適合していないことがわかる。

次に, モデル B において精度は 0.971, 再現率は 0.976 および F1 スコアは 0.972 であった。また, モデル B における各キーの入力の混同行列を図 5.13, 学習時における学習曲線を図 5.10 に示す。モデル A と同様に学習の進行に問題はなく, かつ過剰適合を起こしていないことがわかる。

また, モデル B を用いて変換した入力領域を図 5.14 を示す。モデル B によって各キーの入力領域は広くなるように変換されるが, キーのサイズを大きくするような単純な変換ではなく, キーごとに入力領域が調整されていることがわかる。ここで, モデル A は座標の位置以外にも入力にホバーの情報が必要になるため, 図 5.14 のような入力領域を示した図を作成で

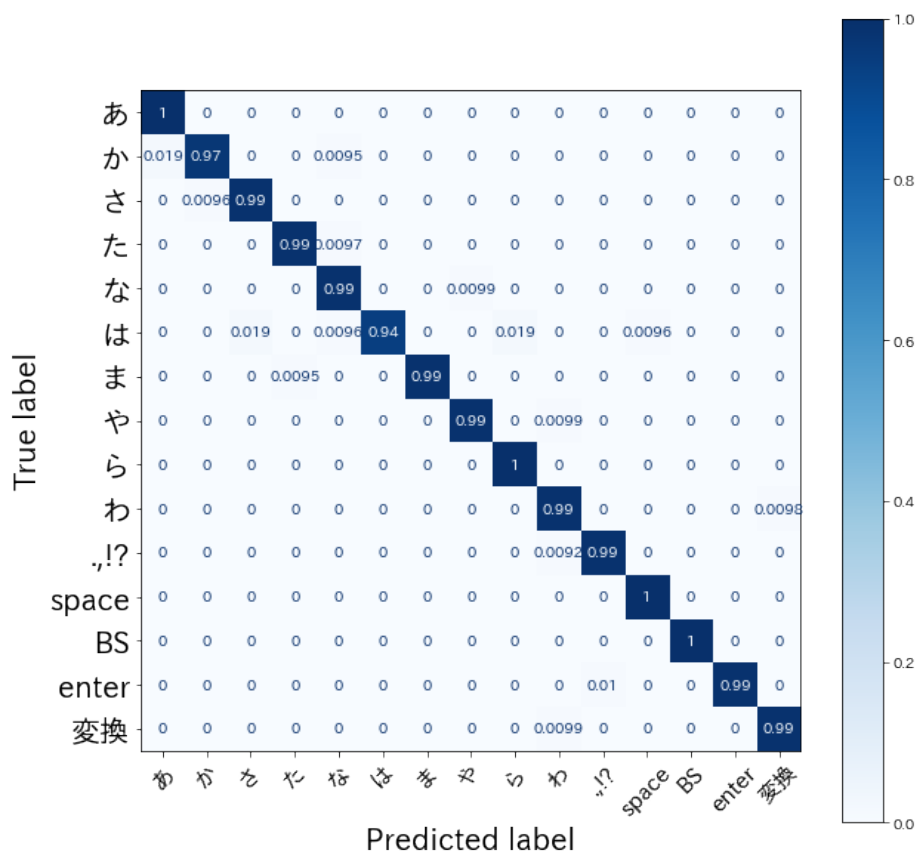


図 5.13: モデル B の混合行列.

きない.

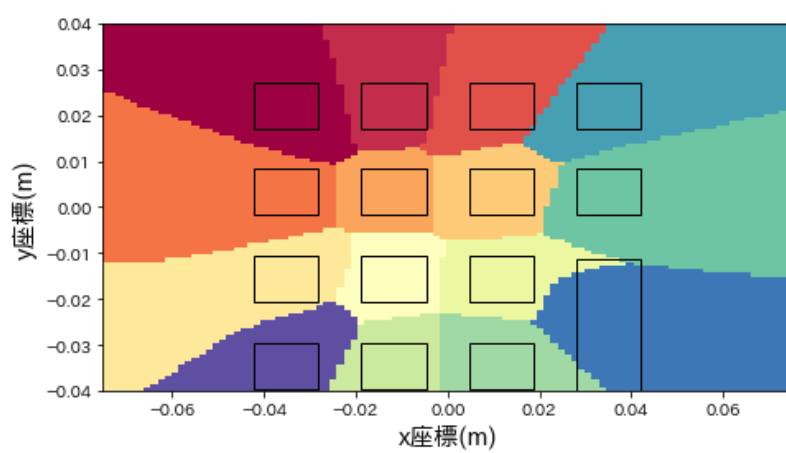


図 5.14: モデル B を用いて変換した入力領域. 各色は一つのキーの入力領域を示す. また四角の黒線はキーの表示位置を示す.

第6章 比較実験

本手法および入力支援による文字入力性能と従来の VR 内におけるフリック入力手法の文字入力性能の比較実験を行う。入力性能の評価には入力速度と入力精度を使う。ここで、入力速度を 1 分間当たりの入力した文字数を示す Characters Per Minute（以降、CPM）とする。また、エラー率をすべての入力に対する誤入力の割合を示す Total Error Rate（以降、TER）[23] とする。

6.1 比較手法

比較する入力手法は、タッチ位置のみを示す手法（以降、手法 1）、3.3.1 において示したホバー位置を示すポインタを付した入力手法（以降、手法 2）、予備実験において作成したモデル A を用いて調整した入力領域を用いた入力手法（以降、手法 3）、モデル B を用いて調整した入力領域を用いた入力手法（以降、手法 4）および従来の入力手法（以降、従来手法）である。従来手法には福仲ら [6] の手法を参考に作成した VR 内におけるフリック入力手法を用いる。

6.2 手順

この実験は著者 1 人によって行う。参加者は HMD を装着し、椅子に座った状態において右手を用いて 27 個の短文（計 213 文字）からなる課題文を高速、かつ正確に入力する入力タスクを行う。課題文には、50 音節、濁音、半濁音、送り仮名、長母音からなる意味を持つ 5～11 文字の短文（表 A.1）を用いる。

各手法間における実験結果の偏りを抑えるために 5 つの手法において、入力タスクを各 3 回行う。この時、各手法の実行順は 15 回の入力タスクにおいて、ランダムである。それぞれの入力タスクの課題文のセットにはすべての試行において同じものを使用する。入力タスクを終えた後、参加者は次の入力タスクを始めるまで 10 分以上の休憩をとる。

6.3 実験結果

この節において、実験において得られた各手法の入力速度（CPM）を図 6.1 に、また、エラー率（TER）を図 6.2 に示す。

6.3.1 入力速度 (CPM)

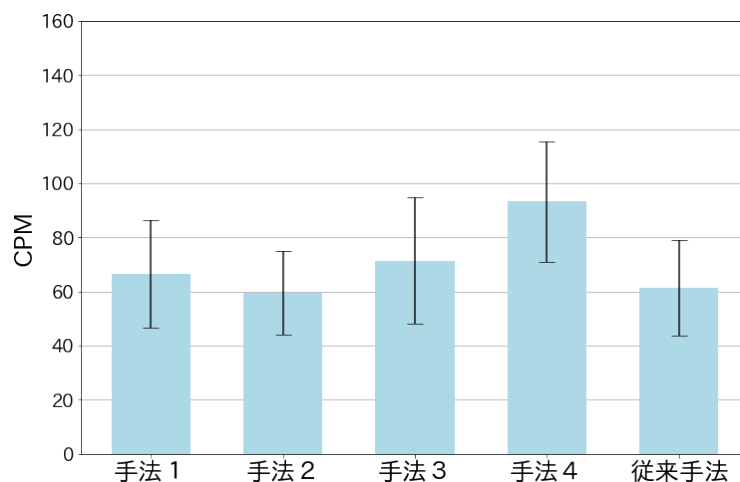


図 6.1: 各手法の入力速度 (CPM) の平均を示した棒グラフ。エラーバーは標準偏差を示す。

各手法における入力速度 (CPM) を図 6.1 に示す。それぞれの手法における入力速度 (CPM) は、手法 1 が平均 66.4 ($SD=19.9$)、手法 2 が平均 59.5 ($SD=15.4$)、手法 3 が平均 71.5 ($SD=23.3$)、手法 4 が平均 93.3 ($SD=22.3$) および従来手法が平均 61.4 ($SD=17.8$) であった。

6.3.2 エラー率 (TER)

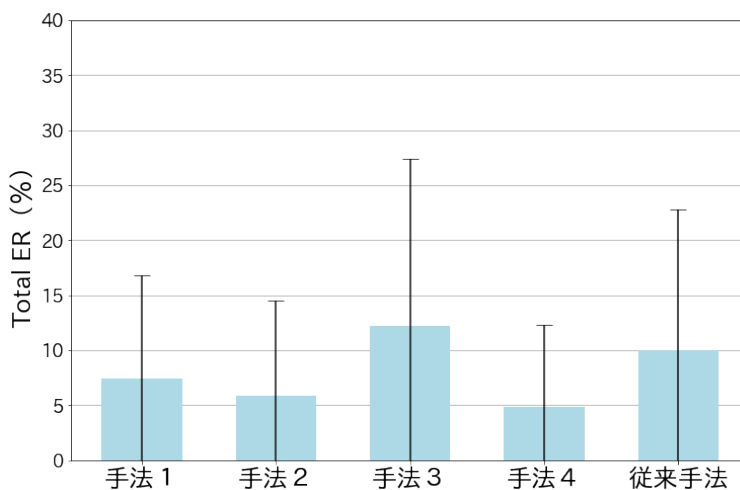


図 6.2: 各手法のエラー率 (TER) の平均を示した棒グラフ。エラーバーは標準偏差を示す。

各手法におけるエラー率 (TER) を図 6.2 に示す. 手法 1 は平均 7.5 % ($SD=9.3$ %), 手法 2 は平均 5.9 % ($SD=8.7$ %), モデル 3 は平均 12.2 % ($SD=15.2$ %), 手法 4 は平均 4.9 % ($SD=7.4$ %) および従来手法は平均 10.0 % ($SD=12.8$ %) であった.

6.4 考察

図 6.1 より, 手法 4 は手法 1 および手法 3 を含む他の入力手法より入力速度が速いことがわかる. このことから, 入力位置のみによる入力領域の調整は入力速度の向上に有効であると考えられる. また, 図 6.2 より, 手法 3 は他の手法よりも TER が高いこと, および手法 1 と手法 2 の TER は同程度であることがわかる. このことから, 指のホバー位置を取り入れた入力領域の調整およびホバー位置の表示による支援には入力精度を向上させる効果がないと考えられる.

第7章 議論および今後の課題

本章においては、評価実験の結果に基づいて提案手法の問題点および改善案について議論し、今後の課題を述べる。

7.1 複数人による実験

今回は本手法の文字入力性能の評価実験を著者1人のみによって行った。しかし、参加者が1人のみ、かつ、提案手法の開発過程において提案手法の操作に慣れている著者による結果は偏りが生じるため一般的な結果ではない。そのため、結果の信頼性を高めるために、複数人による実験を行う必要がある。

7.2 入力領域を調整する機械学習モデルの改良

今回は入力領域の変換を行う際に深層学習モデルを用いた。しかし、入力は12値であり、出力も触れた点を表すx,y座標の2値のみであるため、深層学習のような複雑なモデルを用いずとも行うことができる。今回はUnityにおいて機械学習モデルを動作させるために、PyTorchによって深層学習モデルを作成したが、それ以外の方法によって、Unityにおいて入力領域の変換を行うことができれば、必ずしも深層学習モデルを用いる必要はない。

7.3 学習に用いるデータの数

今回の実装において、入力領域を変換するモデルの学習のために、著者の入力のデータを1500個用意した。しかし、ユーザ1人あたりに対して1500個のデータを収集することは長い時間を必要とするため負担がかかる。そこで、学習に必要なデータ量と入力精度の関係を調べることで、入力精度を維持しつつデータ収集時にユーザの負担にならない適切なデータ量を検討する。

7.4 キーの大きさ

今回は、入力位置の信頼楕円をもとに誤入力が少なくなるようにキーを配置し直した。しかし、キーの大きさが入力に与える影響については未検証である。今回用いたモデルによる

入力領域の変換は図 5.14 に示したが、これによると、エンターキーを除いたキーの本来の入力領域は変換した入力領域の範囲内に収まっており、入力領域が著しく変わることはなかった。ここで、単にキーのサイズを大きくすることによって入力領域を変換するモデルを用いずとも、今回実装した提案手法と同程度の文字入力を行うことができるか調査する必要がある。

第8章 結論

本研究においては，HMDの前面に取り付けたタッチデバイスを用いてフリック入力を行うVR用のフリック文字入力手法を示した．また，本手法における入力性能を向上させるための入力支援を考案して，それぞれにおける文字入力性能を実証する比較実験を行った．この実験の結果，モデルBを用いたフリック入力は従来のVR用フリック文字入力手法と比べて入力速度が大きいことおよびエラー率が小さいことを示した．今後は，複数人による実験を行って本手法の文字入力性能を調査する．

謝辞

本研究を進めるにあたって、一年を通して志築文太郎先生にはご意見とご指導をいただきました。心から感謝いたします。研究室の先輩、および同輩の方々、特に西川宜利氏には一年間を通して研究の方針の相談、論文執筆、および添削など多大な支援をいただきました。

最後に、学生生活においてお世話になった教授、友人、また私の生活を支えてくれた両親に深く感謝申し上げます。

参考文献

- [1] 日本バーチャルリアリティ学会. 日本バーチャルリアリティ学会 バーチャルリアリティとは. <https://vrsj.org/about/virtualreality/>. (最終閲覧日 2023 年 2 月 1 日) .
- [2] Mark R. Mine. Virtual Environment Interaction Techniques. USA, 1995. University of North Carolina at Chapel Hill.
- [3] ultraleap. Tracking — VR Developer Mount — Ultraleap. <https://www.ultraleap.com/product/vr-developer-mount/>. (最終閲覧日 2023 年 2 月 1 日) .
- [4] 長澤直子. 大学生のスマートフォンと PC での文字入力方法. コンピュータ&エデュケーション, Vol. 43, pp. 67–72, 2017.
- [5] 喜多修太郎, 小倉加奈代, Bista Bhed Bahadur, 高田豊雄. LeapMotion を用いた VR 上での文字入力手法の検討. 情報処理学会研究報告, Vol. 2019-HCI-181, No. 21, pp. 1–7, 2019.
- [6] 福仲伊織, 謝浩然, 宮田一乗. VR 環境におけるフリック入力形式インタフェースの開発. 情報処理学会研究報告, Vol. 2019-HCI-182, No. 3, pp. 1–8, 2019.
- [7] 小澤宗馬, 梅澤猛, 大澤範高. J-036 空中におけるつまむ動作を用いた効率的な文字入力の検討 (J 分野: ヒューマンコミュニケーション&インタラクション, 一般論文). 情報科学技術フォーラム講演論文集, Vol. 14, No. 3, pp. 389–390, 2015.
- [8] Kenneth R. Boff, Lloyd Kaufman, and James P. Thomas. Handbook of perception and human performance, Vol. 1. Wiley New York, 1986.
- [9] 川口航平, 磯本俊弥, 志築文太郎, 高橋伸. VR 向けの掌上における日本語フリック入力手法の提案. ヒューマンインターフェースシンポジウム 2019 論文集, pp. 676–682, 2019.
- [10] R. William Soukoreff and I. Scott MacKenzie. Recent Developments in Text-Entry Error Rate Measurement. In CHI '04 Extended Abstracts on Human Factors in Computing Systems, CHI EA '04, pp. 1425–1428, New York, NY, USA, 2004. Association for Computing Machinery.
- [11] Jay Henderson, Jessy Ceha, and Edward Lank. STAT: Subtle Typing Around the Thigh for Head-Mounted Displays. In 22nd International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI '20, New York, NY, USA, 2020. Association for Computing Machinery.

- [12] Rio Hirai, Ryo Ikeda, and Buntarou Shizuki. Preliminary Investigation of Text Entry Method with Haptic Feedback from Real Object Surfaces Estimated Using Hand Tracking on HMD. In Asian CHI Symposium 2021, Asian CHI Symposium 2021, pp. 129–131, New York, NY, USA, 2021. Association for Computing Machinery.
- [13] Youngwon R. Kim and Gerard J. Kim. HoVR-Type: Smartphone as a Typing Interface in VR Using Hovering. In Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology, VRST '16, pp. 333–334, New York, NY, USA, 2016. Association for Computing Machinery.
- [14] Chien-Min Wu, Chih-Wen Hsu, Tzu-Kuei Lee, and Shana Smith. A Virtual Reality Keyboard with Realistic Haptic Feedback in a Fully Immersive Virtual Environment. Virtual Reality, Vol. 21, No. 1, pp. 19–29, 2017.
- [15] Jan Gugenheimer, David Dobbstein, Christian Winkler, Gabriel Haas, and Enrico Rukzio. FaceTouch: Enabling Touch Interaction in Display Fixed UIs for Mobile Virtual Reality. In Proceedings of the 29th Annual Symposium on User Interface Software and Technology, UIST '16, pp. 49–60, New York, NY, USA, 2016. Association for Computing Machinery.
- [16] Jihyun Lee, Byungmoon Kim, Bongwon Suh, and Eunye Koh. Exploring the Front Touch Interface for Virtual Reality Headsets. In Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems, CHI EA '16, pp. 2585–2591, New York, NY, USA, 2016. Association for Computing Machinery.
- [17] Wahyu Hutama, Hikari Harashima, Hironori Ishikawa, and Hiroyuki Manabe. HMK: Head-Mounted-Keyboard for Text Input in Virtual or Augmented Reality. In Adjunct Proceedings of the 34th Annual ACM Symposium on User Interface Software and Technology, UIST '21 Adjunct, pp. 115–117, New York, NY, USA, 2021. Association for Computing Machinery.
- [18] Wen-Jie Tseng, Li-Yang Wang, and Liwei Chan. FaceWidgets: Exploring Tangible Interaction on Face with Head-Mounted Displays. In Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology, UIST '19, pp. 417–427, New York, NY, USA, 2019. Association for Computing Machinery.
- [19] Masao Ito. Movement and thought: identical control mechanisms by the cerebellum. Trends in Neurosciences, Vol. 16, No. 11, pp. 448–450, 1993.
- [20] Katrin Wolf, Christian Müller-Tomfelde, Kelvin Cheng, and Ina Wechsung. Does Proprioception Guide Back-of-Device Pointing as Well as Vision? In CHI '12 Extended Abstracts on Human Factors in Computing Systems, CHI EA '12, pp. 1739–1744, New York, NY, USA, 2012. Association for Computing Machinery.

- [21] Tayfur Coskun, Christoph Bruns, Amal Benzina, Manuel Huber, Patrick Maier, Marcus Tönnis, and Gudrun Klinker. Gestyboard BackTouch 1.0: Two-Handed Backside Blind-Typing on Mobile Touch-Sensitive Surfaces. In Ivan Stojmenovic, Zixue Cheng, and Song Guo, editors, Mobile and Ubiquitous Systems: Computing, Networking, and Services, pp. 422–434, Cham, 2014. Springer International Publishing.
- [22] 井川洋平, 宮下芳明. アイズフリーで速記できる「方向のみ」のフリック入力手法. インタラクション 2013 論文集, pp. 651–656, 2013.
- [23] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. Analysis of text entry performance metrics. In 2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH), pp. 100–105, 2009.

著者論文リスト

本論文に関する論文

- 査読なし国内会議論文

1. 山口 泰生, 西川 宜利, 志築文太郎. HMD 前面を用いた VR 用フリック文字入力手法の提案. 第 30 回インタラクティブシステムとソフトウェアに関するワークショップ, 日本ソフトウェア科学会, 2022, 3 pages.

その他の論文

- 査読なし国内会議論文

1. 平井李音, 鈴木健介, 山口泰生, 志築文太郎. VR 環境向けの焦点座標を用いた 3 次元バブルカーソルの評価. 第 30 回インタラクティブシステムとソフトウェアに関するワークショップ, 日本ソフトウェア科学会, 2022, 2 pages.

付録A 実験において用いた課題文

表 A.1: 実験において用いた課題文

漢字を含む文	実験参加者が入力する文
思い切りも必要	おもいきりもひつよう
今日も徹夜	きょうもてつや
筑波山に登る	つくばさんにのぼる
つくばエクスプレス	つくばえくすぷれす
メールを送信	めーるをそうしん
これもお願い	これもおねがい
続きが気になる	つづきがきになる
みんなでミニゲーム	みんなでみにげーむ
プログラミング	ぷろぐらみんぐ
発表会に臨む	はっぴょうかいにのぞむ
ページをめくる	ページをめくる
仙台へ旅行	せんだいへりょこう
百円玉	ひゃくえんだま
やっぱりいいよ	やっぱりいいよ
水を補充する	みずをほじゅうする
鍋奉行をする	なべぶぎょうをする
パソコンは高い	ばそこんはたかい
ありがとうございます	ありがとうございます
全国都道府県	ぜんこくとどうふけん
記録を縮める	きろくをちぢめる
僅かにとどまる	わずかにとどまる
マット運動が得意	まっとうんどうがとくい
湯冷めした	ゆざめした
足跡をたどる	あしあとをたどる
シャワーを浴びた	しゃわーをあびた
お湯がぬるくなる	おゆがぬるくなる
犬と散歩	いぬとさんぽ