修士論文

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

1710273　福仲　伊織

主指導教員　宮田　一乘
審査委員主査　宮田　一乘
審査委員　吉高　淳夫
赤木　正人
小谷　一孔

北陸先端科学技術大学院大学
先端科学技術研究科
(情報科学)

平成 31 年 2 月

目 次

図 目 次

表 目 次

第1章　序論

本章では，はじめに研究背景について述べる．つづいて，研究の目的について述べ，最後に本論文の構成について示す．

1.1　研究背景

HTC VIVE などの HMD(Head Mounted Display) の登場により，VR（Virtual Reality）は一般的に普及してきた．さらには，近年，Unity3D[1]などの VR 開発環境が整ってきたことで，VR を利用した多種多様なコンテンツが登場してきており，今後もさらに VR は普及していくと予想される．

VR コンテンツの体験には HMD が必須であるが，HMD を装着することによるさまざまな制限があり，その典型的な例は，視界が遮られることによって現実空間の把握が難しくなることである．そのため，文字入力を行う際には物理キーボードの位置を確認できず，物理キーボードを利用した文字入力は非常に難しい．現在，VR における文字入力手法として，図 1.1 のようにコントローラで仮想キーボードのキーを選択して入力する方法が一般的であるが，コントローラを用いて小さなキーを正確に狙うのは難しく，ユーザが自分の手で直接入力する，パーソナルコンピュータのキーボード入力や，スマートフォンやタブレット端末で若者たちを中心に広く利用されているフリック入力などの従来から使用されている入力方法と比較すると，入力速度や誤入力の頻度などの操作性の面で劣る．

VR 空間における文字入力手法の研究はこれまでにもいろいろと行われてきたが，そのほとんどが従来の入力手法とは異なる特殊な入力手法を模索していたり，QWERTY 配列のキーボード入力形式を対象としており，フリック入力に着目した研究は非常に少ない．特殊な入力手法はユーザが操作方法を理解するまでに時間がかかり，素早く正確に入力するためには長時間の練習と慣れが必要となる．また，ユーザは日常的に VR 空間で生活するわけではないので，普段利用する入力手法と VR 空間内で利用する入力手法は同一であったほうが便利である．一方，キーボード入力は日常的に利用されているが，近年日本において，キーボード入力を苦手とする若者が増えており，特にスマートフォンなどの携帯端末において日本語入力を行う際にキーボード入力を利用する若者は非常に少なく，ほとんどがフリック入力を利用している [2]. フリック入力はキーボード入力と比較すると，キー

[1] `https://unity3d.com/jp`(2019.02.19)

の大きさが同じとき，狭い範囲でキーボード全体を表示することができることや，キーボードでローマ字入力をするときに比べ，キー入力回数の少なさから高速に日本語入力が行えることが利点として挙げられる．

また，近年では，HTC VIVE Pro[2]のようにユーザの手指をトラッキングすることのできる HMD を利用したり，Leap Motion[3]などの手指をトラッキングできる機器と既存の HMD を組み合わせたりするなど，手に機器を保持せずとも，ユーザの手指の動きを VR 空間に持ちこんで仮想物体を直接的に操作する手法が確立されてきている．このように，フリーハンドで体験できるコンテンツが登場し始めており，文字入力のためだけにコントローラをわざわざ用意することは理にかなっていない．

以上の背景を踏まえて，本研究では VR 環境においてフリック入力と同様の操作性を持つ文字入力インタフェースの開発を行い，視覚的フィードバックを実装することで操作性の向上を目指す．

(1) VRChat(VRChat Inc.)　(2) AltSpaceVR(AltSpace Inc.)

図 1.1: 現在 VR コンテンツにおいて利用されている文字入力手法の例

1.2　研究目的

本研究の目的は，VR 環境においてフリック入力で日本語ひらがなを入力することができるインタフェースを開発し，視覚的フィードバックを実装することで操作性の向上を目指すことである．また，開発したインタフェースを VR における仮想キーボードを利用した既存の文字入力と，入力速度および誤入力の頻度について比較し有効性を評価する．本研究で作成したインタフェースを利用することで，フリーハンド VR 環境においても，日常的に利用するフリック入力と同じ感覚で日本語ひらがなが入力できるようになることを目指す．

[2] https://www.vive.com/jp/product/vive-pro/(2019.02.19)
[3] https://www.leapmotion.com/ja/(2019.02.19)

1.3　本論文の構成

本論文は，全 5 章で構成する．第 2 章では，関連研究として Leap Motion を用いた文字入力手法について述べ，本研究の位置づけを明らかにする．次に第 3 章で本研究で開発したインタフェースについて詳述し，第 4 章で本研究で開発したインタフェースの評価実験の結果と考察を示す．最後に，第 5 章で本研究を総括し，今後の課題について述べる．

第2章　関連研究

本章では，フリック入力，フリーハンド VR 環境を実現するためによく利用される Leap Motion について説明し，その後，Leap Motion を利用した文字入力手法に関する研究について説明したのち，本研究の位置づけについて述べる．

2.1　フリック入力

スマートフォンやタブレット端末などのタッチパネルを備えた機器で用いられる「フリック」とはタッチパネルを指で押圧した後，弾くように動かす操作方法で，この操作によって文字入力を行う入力方法を一般的にフリック入力と呼ぶ．

この文字入力手法は，各行のあ段をテンキー風に配置して，フリックした場所と方向によって入力する文字を決定する (図 2.1)．

図 2.1: フリック入力の様子．(通常時 (左) と入力中 (右))

2.2　Leap Motion

『「Leap Motion」は，米 Leap Motion 社から発売されている「手」や「指」の検出に特化したセンサ』[3] である．開発者向けに，Unity3D で利用できる Leap Motion Core Assets[3]が同社より提供されていたり，HTC VIVE や Occulus Rift といった HMD にとりつけるためのマウンターが提供されていたりするため，VR 環境に現実の手指の動きをトラッキングして持ち込む際によく利用されている．

[3] https://developer.leapmotion.com/unity#5436356(2019.02.19)

図 2.2: Leap Motion

2.3 Leap Motion を用いた文字入力手法

細野ら [4] は，両手をかざすことで指先を中心として円環状に右手に子音を左手に母音を表示し，指を動かすことによって濁音半濁音を含まないひらがな文字を入力する手法を提案した (図 2.3). 宮田ら [6] は，ユーザの手が，Grab 状態であるかどうかの判断と指の動きの組み合わせを両手で行うことによってアルファベットを入力する手法を提案した (図 2.4). 二本松ら [7] は，親指とその他の指を接触させるピンチ動作を用いて，アルファベットと特殊文字を接触させる指の種類ごとに 8 種類ずつ円状に表示させ，表示させた文字の上でピンチを解除することによって文字入力を行う手法を提案した (図 2.5).Komiya ら [10] は，両手の指に子音を割り当て，入力したい行の子音に対応する指を曲げ伸ばしすることでその行の五音を指に割り当て，入力したい文字に対応する指を曲げ伸ばしすることでひらがな文字を入力する手法を提案した (図 2.6)．これらの入力手法は PC やスマートフォンなどで一般的に利用されている入力方法とは大きく異なっており，ユーザがすぐに操作方法を理解できるとは限らない．

また，小澤ら [5] は，ひらがなのあ段を一定間隔で配置し，つまむ動作で子音を選択，つまむ動作のまま指を移動させることで文字を選び，指を離すことで文字を入力する手法を提案した (図 2.7)．この入力手法はフリック入力を模してはいるものの，スマートフォンなどで行うフリック入力と全く同じ動作で行える手法ではない．

他にも，一般的に利用されている文字入力手法を模したものとして，VR Hex Keyboard by JCorvinus[4](図 2.8) のように QWERTY 配列のキーボードを直接タイプして入力が行えるものも開発されている．

一方，喜多ら [8] は，既存入力手法である，キーボード入力とフリック入力を VR 環境で実装し，どちらのほうが VR 環境に適しているのかを調べた (図 2.9)．この

[4] https://github.com/jcorvinus/VRKeyboard(2019.02.19)

研究では，フリック入力はキーボード入力と比較するとVR環境での文字入力に適していないとされているが，この研究で行われたアンケート調査によると，回答者の多くがVR環境の文字入力の際にフリック入力を利用したいと回答しており，一定の需要があるとされている．また，この研究で使用されたキーボードにはフィードバックは実装されておらず，その効果は不明である．

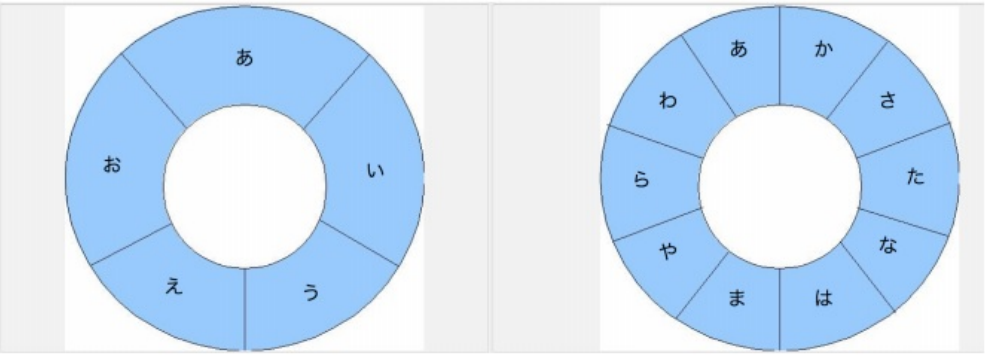


図 2.3: 細野らが提案した文字入力画面 [4]

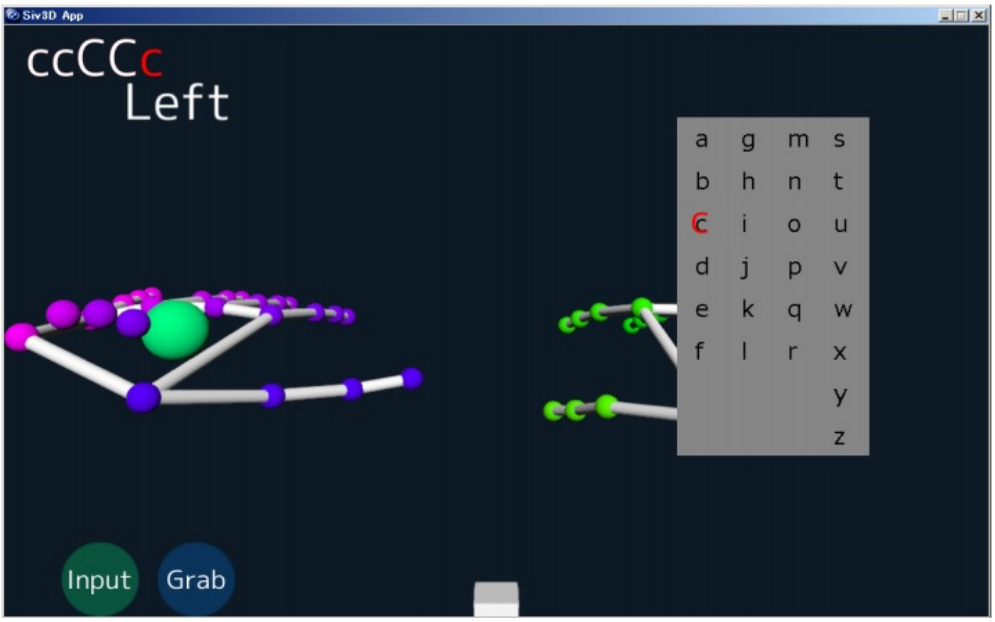


図 2.4: 宮田らが提案した文字入力画面 [6]

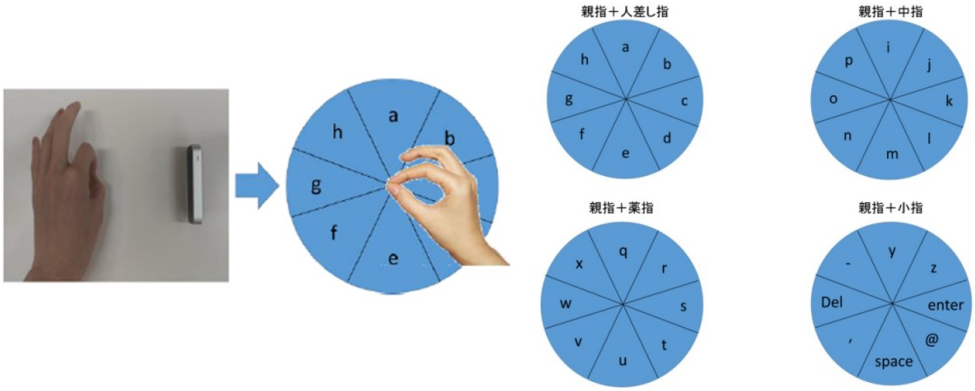


図 2.5: 二本松らが提案した文字入力画面 [7]

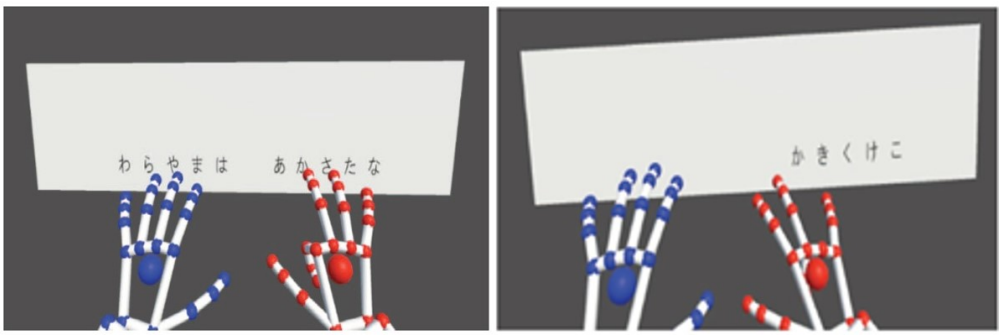


図 2.6: Komiya らが提案した文字入力画面 [10]

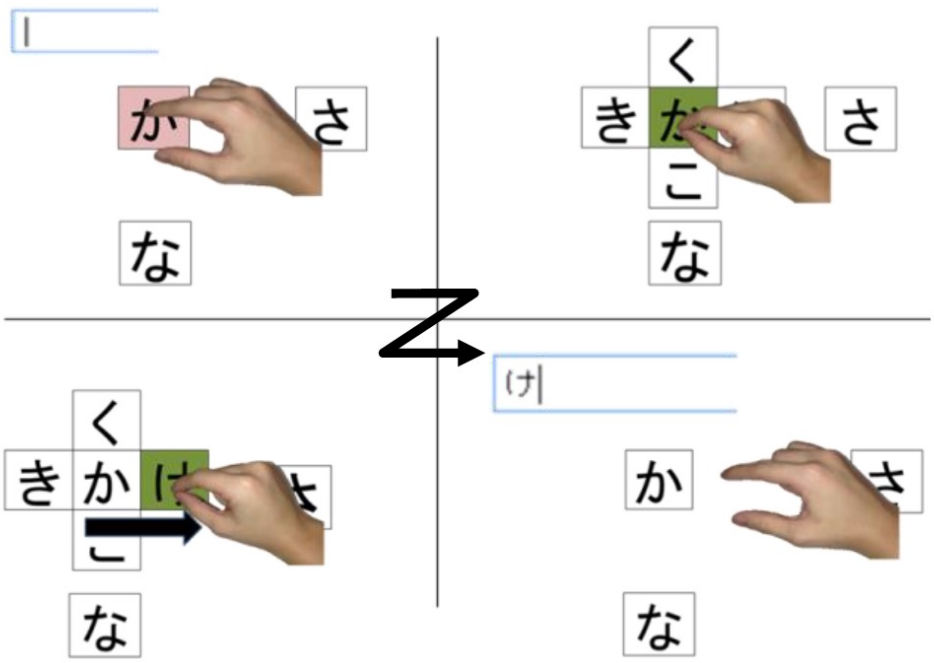


図 2.7: 小澤らが提案した文字入力画面 [5]

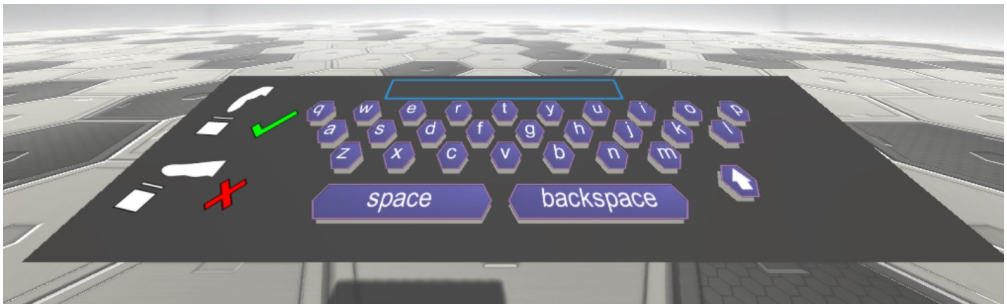


図 2.8: JCorvinus が開発した仮想キーボード

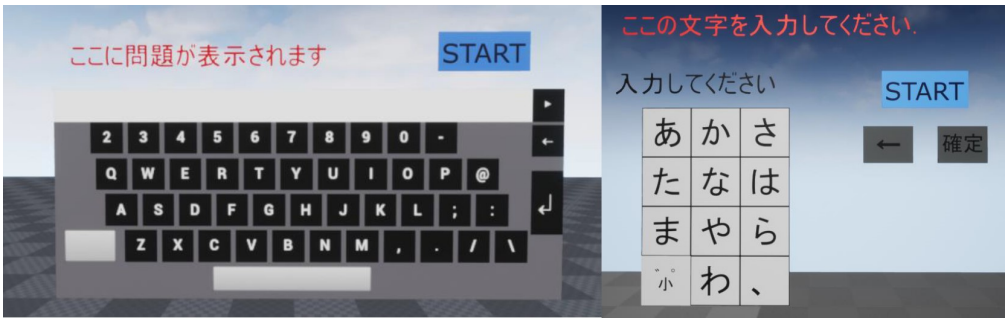


図 2.9: 喜多らが実装した仮想キーボード [8]

2.4　本研究の位置づけ

スマートフォンやタブレット端末で利用されているフリック入力では，主に視覚的なフィードバックによって入力の補助をしている．これはタッチパネルは，物理キーボードのように実際にキーを押し込むわけではないので，ユーザがキーを押したかどうかが分かりづらいという問題を解決するためである．フリーハンドのVR 環境で文字入力を行う際にも同様の問題があるのは明らかで，何らかのフィードバックは必要であると考えられる．しかしながら，VR 環境でスマートフォンなどの携帯端末で利用されるフリック入力と同じ操作で利用できる入力インタフェースは非常に少なく，フィードバックを実装しているものに関しては筆者の知る限りでは存在しない．

本研究では，スマートフォンやタブレット端末などで利用される一般的なフリック入力をフリーハンドの VR 環境に持ち込み，視覚的フィードバックを加えることによって入力速度や誤入力の頻度の低下といった操作性の向上を目指す．手指をトラッキングして VR 環境に持ち込むことに関しては，前節の研究を踏襲し，HMD に Leap Motion を装着し利用することで実現する．

第3章　VR環境におけるフリック入力形式インタフェース

本章では，はじめに開発に利用したゲームエンジン Unity3D について述べ，その後，本研究で作成したインタフェースについて説明する．

3.1　Unity3D

Unity3D は，Unity Technologies が開発した統合開発環境を内蔵するゲームエンジンである (図 3.1)．ゲームエンジンであるが，コンピュータゲーム以外のソフトウェア制作でも使用されている．また，Unity3D 内のアセットストアなどから様々なプラグインを追加して使うことができる．本研究では，Unity2018.2.15f1 Personal を利用して開発を行った．

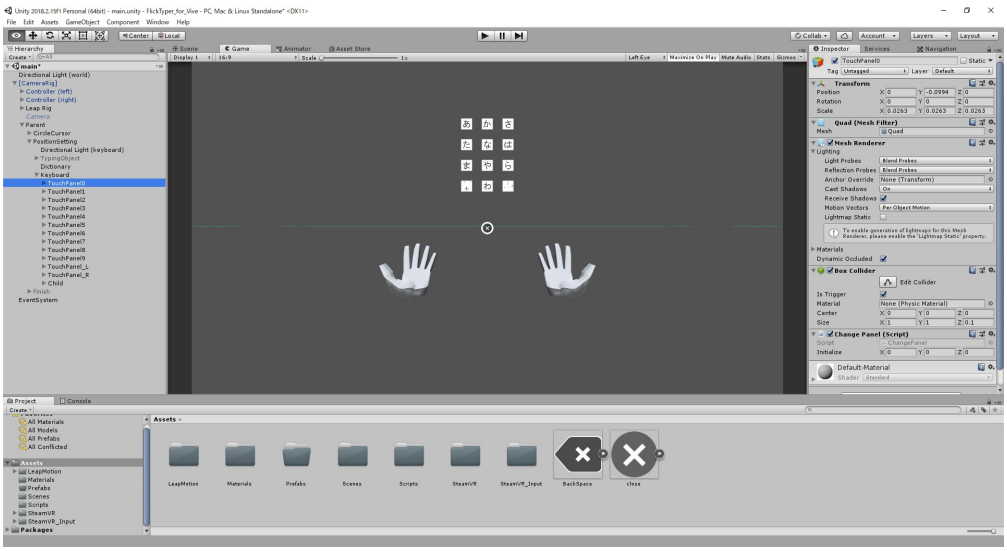


図 3.1: ゲームエンジン Unity3D

3.2　概要

本研究では，Unity3D を用いて，VR 環境においてひらがなで五十音 46 文字と濁音 15 文字，半濁音 5 文字，捨て仮名 10 文字の合計 71 文字をフリック入力と同様の操作で入力ができるインタフェースを開発し，視覚的フィードバックを付与する．作成したインタフェースの外観を図 3.2 に，システムがキーが押されてから入力する文字を確定するまでの流れを図 3.3 に示す．

ユーザが本研究で作成したインタフェースで文字入力を行うときには，VR 環境で仮想の手を操作し，タッチパネルを備えた機器でフリック入力を行うときと同様に，入力したい文字の行のあ段の表示されているキーを押下し，入力したい文字の方向に指を移動させてから指をキーの初期位置よりも手前側に移動させることで入力する文字を確定する．

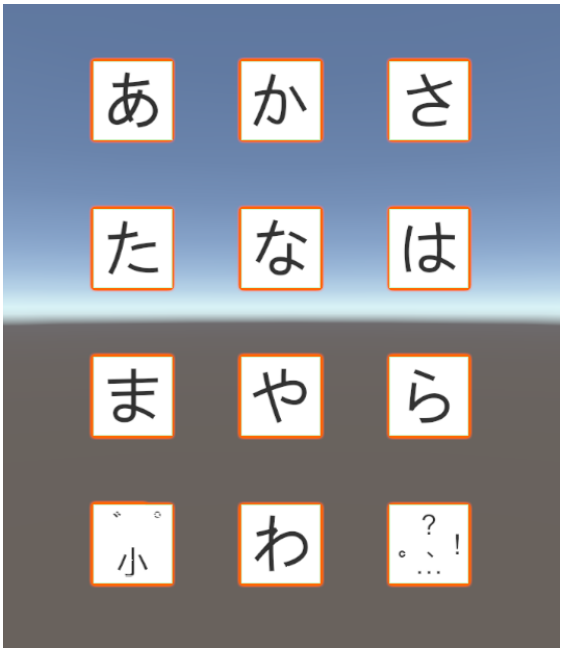


図 3.2: 全体の外観

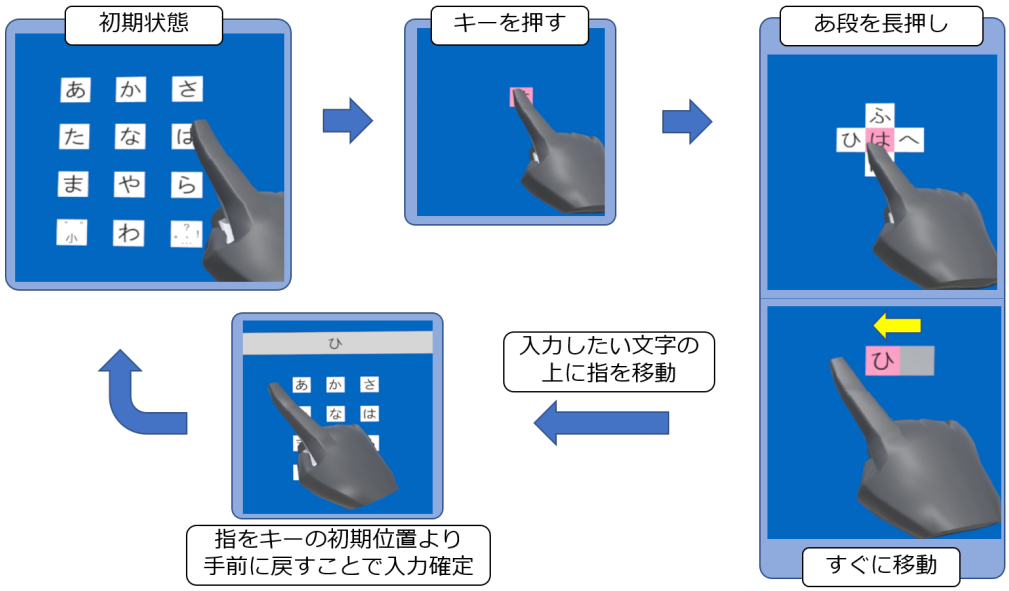


図 3.3: システムの入力確定までの流れ

3.3 インタフェースの実装

本研究で作成したインタフェースの詳細について述べる．

3.3.1 キー

キーに利用しているオブジェクトは Unity3D に標準で用意されている 3D オブジェクトである Quad である．Quad に当たり判定を付与するコンポーネントである Box Collider を図 3.4 のようにアタッチすることで仮想の手とキーとの衝突を検出できるようにする．このキーを 12 個，図 3.2 のようにテンキー風に配置する．

小澤らの研究 [5] において，仮想キーボードのキーの大きさと間隔について『キーの大きさについては 60 ないし 90pixel，キーの間隔については 40 ないし 80pixel であれば適切である』とされている．小澤らの研究では，実験に解像度 1600 × 1200 の 23 インチの液晶ディスプレイを用いているので，ディスプレイに映るキーの大きさ 90pixel と間隔 80pixel を Unity3D 上でオブジェクトとして再現できるようにメートル表記に変換すると，それぞれ，約 26.3 mm と約 23.4 mm となる．本研究ではキーの大きさと間隔について，この値をそれぞれ採用した (図 3.5)．

また，親指と人差し指でつまむ動作によってインタフェース全体を移動できるようにし，ユーザ自身で入力の行いやすい位置にインタフェースを配置できるようにした．移動中は HMD の向きを取得し，インタフェースが視線方向と垂直になるようになっている (図 3.6)．

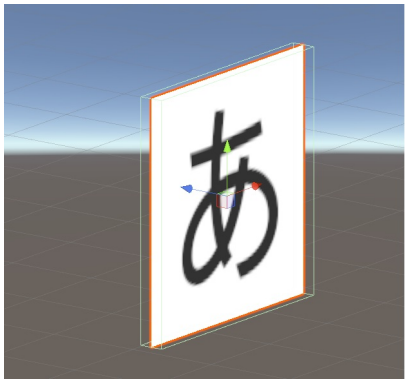


図 3.4: キーの仕組み

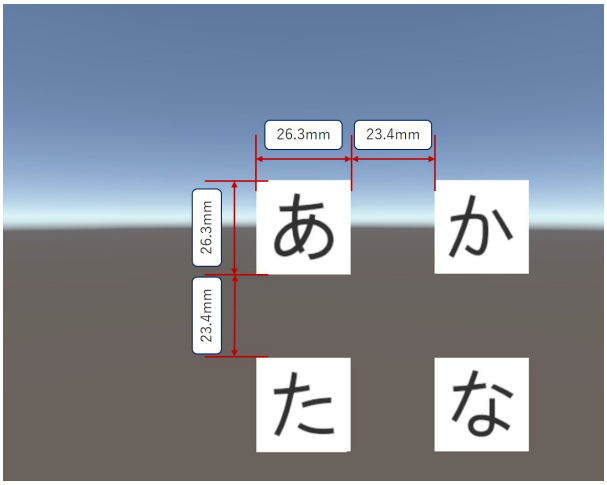


図 3.5: キーの大きさと間隔

図 3.6: インタフェースの移動中の様子

3.3.2　入力

図 3.2 の 12 個のキーのうち，あ段が表示されているキーは押下された後，指先がキーの初期位置よりも手前に移動したときに文字を確定する．入力の判定は，あ段のキーのどれが選択されていたかと，図 3.7 に示す，5 つの領域のどこに指先があるかで判定を行う．領域 0 の範囲はあ段のキーの範囲と一致しており，領域 1〜領域 4 に関しては，領域 0 を含む平面の横軸を x 軸，縦軸を y 軸としたときに，領域 0 を除く範囲を $x+y=0$ または $x-y=0$ を満たす平面で分割した領域である．

一方，最下段左側のキーは押下されると，1 文字前に入力した文字を取得する．清音のひらがなを取得した場合，濁音になれば濁音に，捨て仮名になれば捨て仮名にする．1 文字前に取得した文字が捨て仮名の場合，もとの清音のひらがなが濁音になれば濁音にし，ならないときは清音に戻す．濁音を取得した場合，もとの清音のひらがなが半濁音になれば半濁音にし，ならないときは清音に戻す．半濁音を取得した場合，清音に戻す．

あ段を表示するキー 10 個と濁点，半濁点，小文字を表示するキー 1 個の 11 個のキーを用いることで目標としている 71 文字を入力することができる．や行の領域 1 と 3，わ行の領域 3 と 4，最下段右側のキーの 5 領域については本研究の対象外であるが，Google 日本語入力[5]を参考に，「,」，ー，〜,、,。, ?, ！を割り当てた．

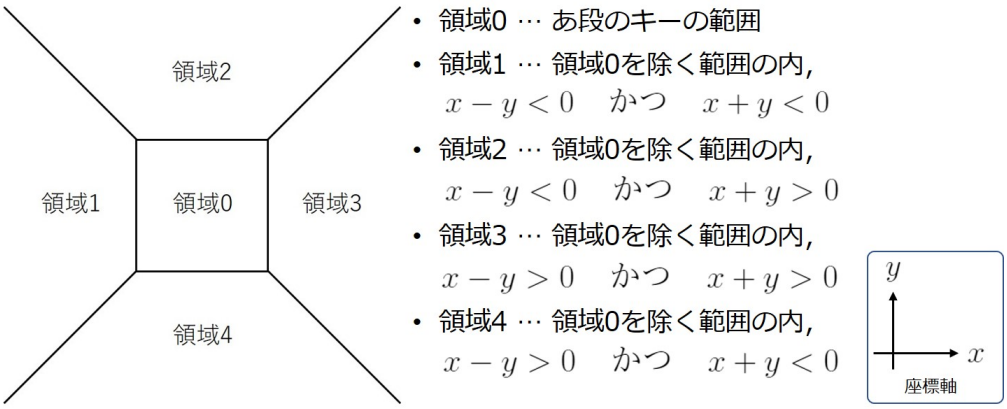


図 3.7: 領域分割

3.4　視覚的フィードバック

本研究で実装した 3 つの視覚的フィードバックについて説明する．後述するキーの展開，キーの色の変化はスマートフォンなどのフリック入力形式のキーボード

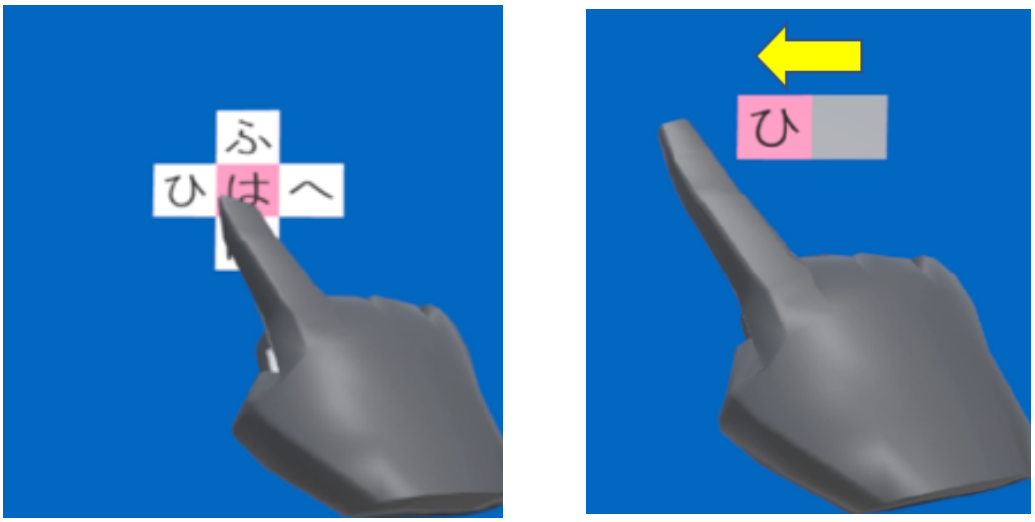
[5]https://www.google.co.jp/ime/(2019.02.21)

に実装されているものを参考にした．実際に利用されているフィードバックを模すことで，スマートフォンなどで入力する感覚に近づけることができるのではないかと考えたためである．また，物理キーボードのように，キーを押し込むことで操作しやすくなるのではないかと考え，後述するキーの押し込み表現を実装している．

他にも，任意の段のキーを操作している間は，他のキーを非表示にすることによって，ユーザがどのキーを操作しているのかを分かりやすくするように設計した．

3.4.1 キーの展開

あ段のキーが押下されると，キーの上下左右の 4 方向にい～お段の文字を表示する Quad を重ならないように配置する．その後，領域 0 にしばらく留まり続けるか，すぐに他の領域に移動するかによって，い～お段の全てを表示するか，指先の移動した領域に対応する段の Quad のみを表示するかを決定する (図 3.8)．



(1) しばらく留まり続けた場合　　(2) すぐに移動した場合

図 3.8: キーの展開

3.4.2　キーの色の変化

あ段のキーが押下されてから入力が確定するまでの間，指先が存在する領域と対応する Quad の色を変化させる．また，前節のキーの展開で一部しか展開されない場合，あ段のキーが選択されていないときにも別の色に変化させている．本研究では図 3.9 に示す色を使用した．

(1) 通常時　　(2) 選択中　　(3) あ段非選択時

図 3.9: 色変化に利用した色．図中にカラーコードを示す．

3.4.3　キーの押し込み表現

あ段のキーが押下されてから入力が確定するまでの間，キーと仮想の手との衝突判定を用いて，キーの初期位置よりも指先が奥側にあるときに，指先に追従してキーが奥行き方向に移動する (図 3.10)．

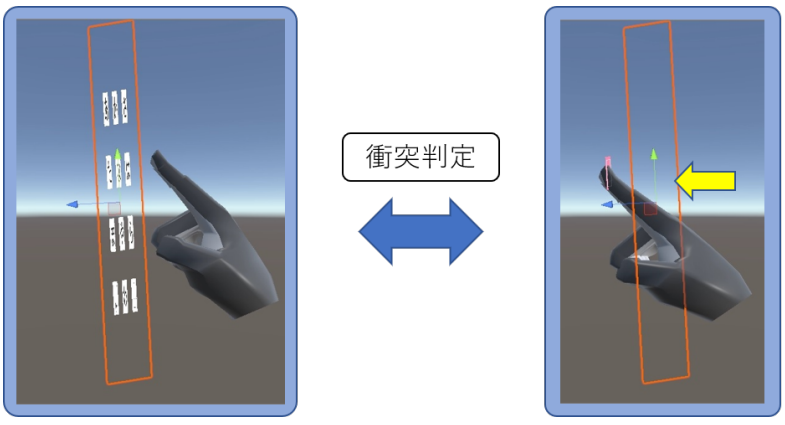


図 3.10: キーの押し込み表現

第4章　実験・評価

本章では，作成したインタフェースの有用性を検証するための評価実験と結果を示す．さらに得られた結果から考察を行う．

4.1　実験環境

実験では，VR を体験するための HMD に HTC VIVE を利用し，Leap Motion は Leap Motion VR Developer Mount[6]を利用して HTC VIVE の前面に取り付け，手指の動きをトラッキングしている (図 4.1)．また，利用した PC の構成は表 4.1 の通りである．

図 4.1: HTC VIVE に Leap Motion を取り付けた様子

[6]https://www.thingiverse.com/thing:445866(2019.02.21)

表 4.1: 使用した PC の構成

OS	Windows 10 Pro(64bit)
CPU	Intel(R)Core(TM)i7-7700 CPU 3.60GHz
GPU	GeForce GTX 1060
RAM	16.0GB

4.2 実験 1　既存の VR 文字入力手法との比較

4.2.1 概要

本研究で作成したインタフェースの有用性を調べるために，既存の VR 文字入力手法との比較を行った．

既存の VR 文字入力手法として，Unity3D のアセットストアからダウンロードできる VRTK-Virtual Reality Toolkit[7]のサンプルシーンに含まれている仮想キーボード (図 4.2) を利用した．本研究で作成したインタフェースと VRTK のキーボードで文字入力を行ってもらい，入力速度と誤入力頻度について比較する．被験者は「本学の日本人学生」とし，内訳は男性 17 名，女性 1 名，いずれも 20 代であった．この 18 名を 9 名ずつの 2 つのグループに分け，グループ 1 には本研究で作成したインタフェース，VRTK の仮想キーボードの順に実験を行ってもらい，グループ 2 にはその逆の順序で実験を行ってもらった．実験には Typing Game System For Unity モジュール[8]を利用して作成した VR 環境のタイピングゲーム (図 4.3) を利用して計測を行った．このタイピングゲームは『START』を押すとゲームが始まり，ランダムに単語が出題される．正しい入力を行うと出題された単語の文字が赤く変化し，単語を最後まで正しく入力すると，次の単語が出題されるという手順を 60 秒間繰り返す．実際に文字入力を行う際，意味のない文字列を入力することは少ないため，出題される単語はランダムな文字列ではなく，日本人であれば誰でも知っていると思われる，日本の都道府県名にした．

1 分間にどれだけ入力できるかという単位は様々あり，KPM(Keys per minutes) や WPM(Words per minutes)，CPM(Characters per minutes) などがよく利用されるが，日本語ひらがな入力に限定すると，キーボード入力において 1key が 1 文字に対応していないことが多いため KPM は入力速度の指標にしづらい．また，WPM については，単語の長さが一様ではないので，一般的に 5 文字を 1word として計測するが，入力した文字数で比較するほうが直接的で分かりやすい．以上の理由から，本研究では入力速度は CPM で計測する．

[7] https://vrtoolkit.readme.io/(2019.02.21)
[8] https://github.com/icoico/TypingGameSystemForUnity(2019.02.21)

本評価実験では，入力した文字数はひらがなで計測する．キーボード入力で入力する文字はアルファベットであるため，ひらがなに変換してから計測した．ひらがなの変換は Typing Game System For Unity モジュールによってキー入力時に行われる．Typing Game System For Unity モジュールによる変換は，訓令式とヘボン式の両方に対応しており，例えば，”si”も”shi”も”し”として変換される．そのため，ユーザは最も慣れた入力順で入力が行える．

また，誤入力頻度は，誤入力回数をキーの入力回数で割った値とする．本システムで入力した文字についてはすべて記録し，どの領域の文字を入力するときに誤入力が多かったかの集計も行う．

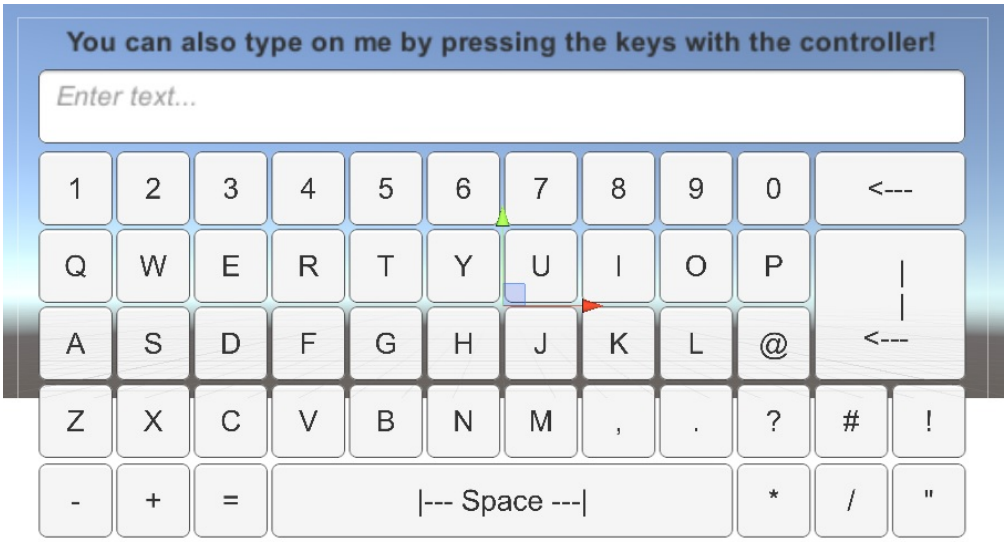
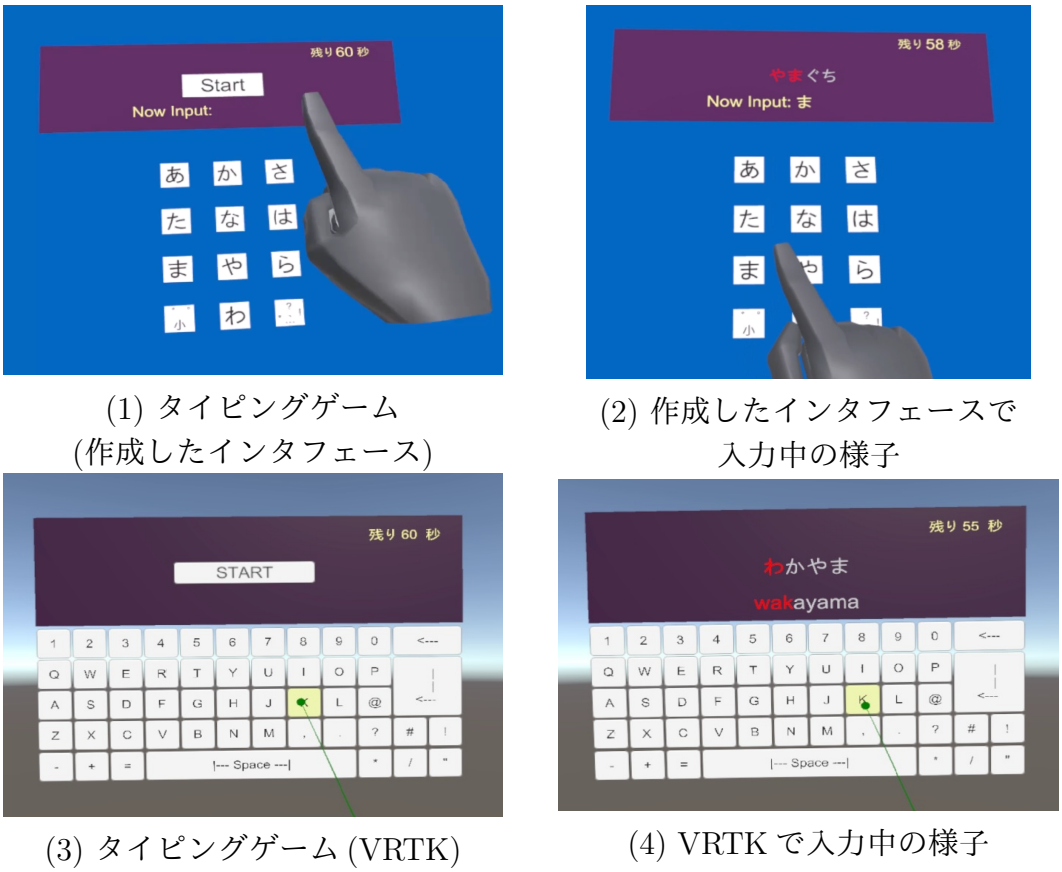


図 4.2: VRTK の仮想キーボード



(1) タイピングゲーム (作成したインタフェース)

(2) 作成したインタフェースで入力中の様子

(3) タイピングゲーム (VRTK)

(4) VRTK で入力中の様子

図 4.3: 実験で利用したタイピングゲーム

4.2.2 実験結果と考察

実験結果を表 4.2 から表 4.5 に示す．また，入力速度，誤入力頻度および領域ごとの誤入力の集計結果をグラフとしてそれぞれ視覚化した結果を図 4.4，図 4.5 に示す. 入力速度について，被験者 18 名分のデータをもとに t 検定を行ったところ，有意な差は見られなかった．また，図 4.4 を見ると，被験者 18 名中 13 名において，既存のコントローラを用いた入力よりも本研究で作成したインタフェースを利用して行った入力の方が素早く入力ができている．このことから既存の VR 文字入力手法と同程度以上の入力速度で入力が行えており，入力速度だけを見ると有用であると考えられる．しかしながら，図 4.5 に示した通り，どの被験者においても，本研究で作成したインタフェースは既存手法と比較して，誤入力頻度が高いという結果になった．単純な平均値での比較だと，本インタフェースを利用した文字入力は，誤入力頻度が既存のコントローラを用いた入力の約 4.6 倍と誤入力頻度の高さが目立つ．

実験の様子を確認すると，仮想の手指の動きがユーザの手指の動きと一致していないことがあり，誤入力の原因のうちのひとつは Leap Motion のトラッキング

の精度によるものであると推測される．一方，誤入力は図 4.6 に示す通り，ほとんどが領域 0 の入力を行う際に生じており，実験の様子を見ると，キーを押し込み，指を引き上げる際に指先が別の領域に移動してしまい，誤入力になっている場合が多い．これは，ユーザがキーを深く押し込みすぎていることが原因と考えられる．すなわち，視覚的フィードバックのために実装している，キーの押し込み表現が誤入力の原因になっている可能性があると推測する．

表 4.2: グループ 1 の実験結果

グループ 1	Flick				QWERTY			
被験者	CPM	キーの入力回数	誤入力回数	誤入力率	CPM	キーの入力回数	誤入力回数	誤入力率
1	24	25	7	28.0%	23	41	0	0.0%
2	55	65	5	7.7%	31	64	4	6.3%
3	63	71	14	19.7%	45	87	4	4.6%
4	37	42	9	21.4%	37	63	3	4.8%
5	55	65	9	13.8%	26	58	8	13.8%
6	26	29	7	24.1%	42	72	4	5.6%
7	14	14	4	28.6%	33	69	5	7.2%
8	73	81	26	32.1%	33	69	13	18.8%
9	72	88	15	17.0%	52	97	4	4.1%

表 4.3: グループ 2 の実験結果

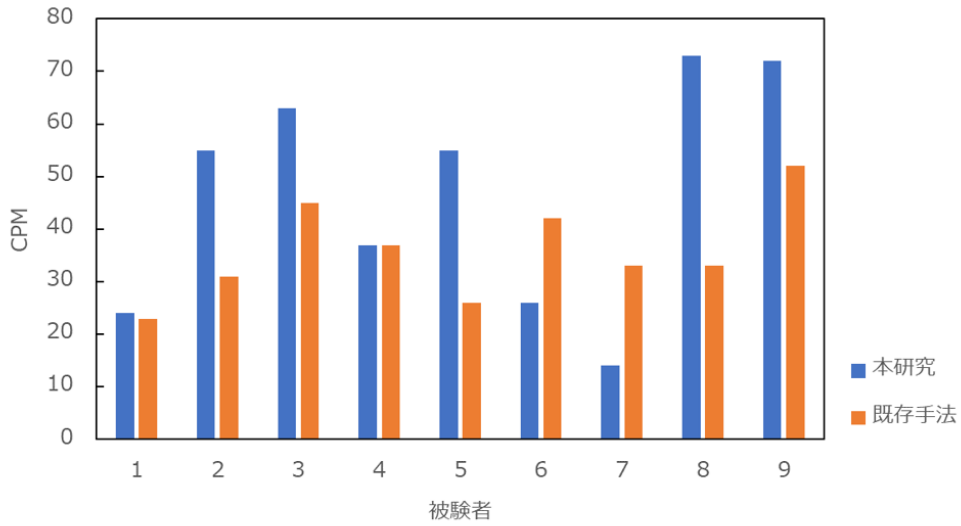
グループ 2	Flick				QWERTY			
被験者	CPM	キーの入力回数	誤入力回数	誤入力率	CPM	キーの入力回数	誤入力回数	誤入力率
10	53	60	11	18.3%	38	66	0	0.0%
11	53	61	20	32.8%	49	95	0	0.0%
12	29	32	13	40.6%	57	106	5	4.7%
13	42	50	4	8.0%	39	74	1	1.4%
14	50	59	6	10.2%	36	69	1	1.4%
15	50	56	32	57.1%	44	82	2	2.4%
16	31	37	2	5.4%	48	88	3	3.4%
17	75	81	12	14.8%	60	115	7	6.1%
18	42	46	5	10.9%	34	63	0	0.0%

表 4.4: グループ 1 の誤入力の集計結果

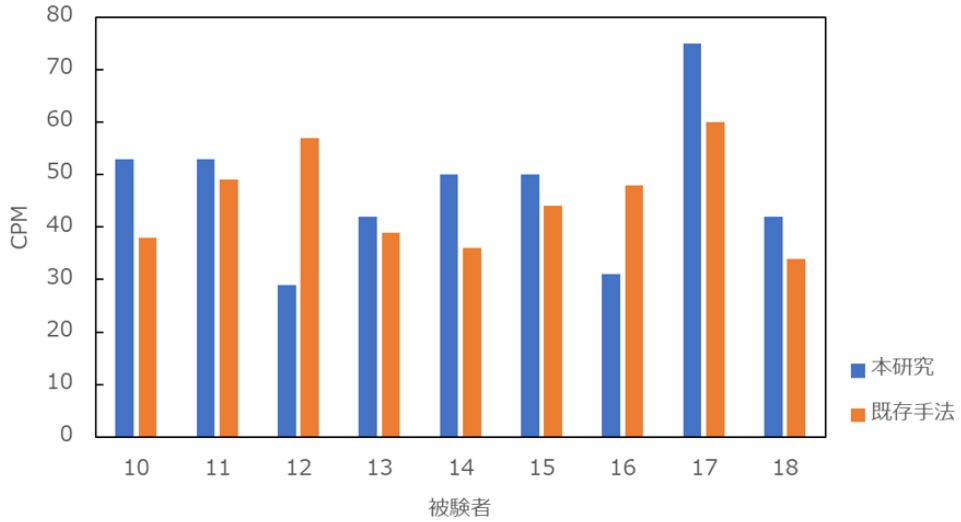
グループ1	領域0	領域1	領域2	領域3	領域4
1	5	1	0	0	1
2	4	1	0	0	0
3	10	3	0	0	1
4	4	1	3	0	1
5	6	0	1	2	0
6	5	2	0	0	0
7	1	3	0	0	0
8	22	1	2	0	1
9	12	2	0	0	1
合計	69	14	6	2	5

表 4.5: グループ 2 の誤入力の集計結果

グループ2	領域0	領域1	領域2	領域3	領域4
10	7	2	0	1	1
11	14	4	0	0	2
12	6	0	6	0	1
13	4	0	0	0	0
14	5	1	0	0	0
15	31	1	0	0	0
16	0	2	0	0	0
17	8	1	0	1	2
18	3	0	2	0	0
合計	78	11	8	2	6

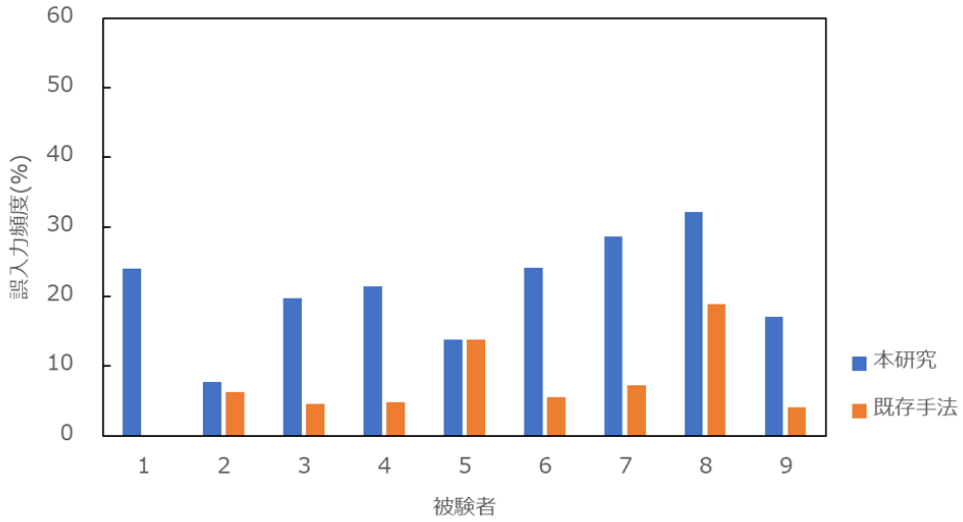


(1) グループ 1 の実験結果

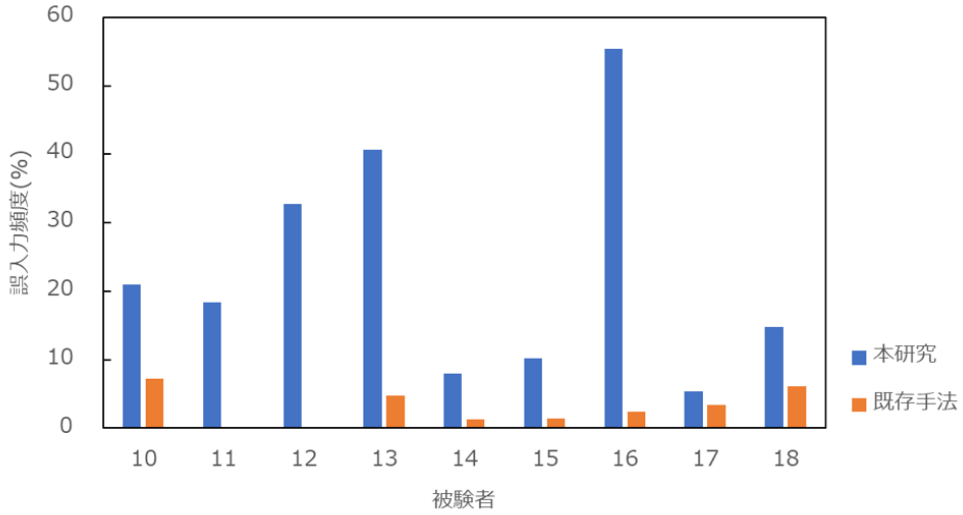


(2) グループ 2 の実験結果

図 4.4: 作成したインタフェースと VRTK との入力速度の比較



(1) グループ 1 の実験結果



(2) グループ 2 の実験結果

図 4.5: 作成したインタフェースと VRTK との誤入力頻度の比較

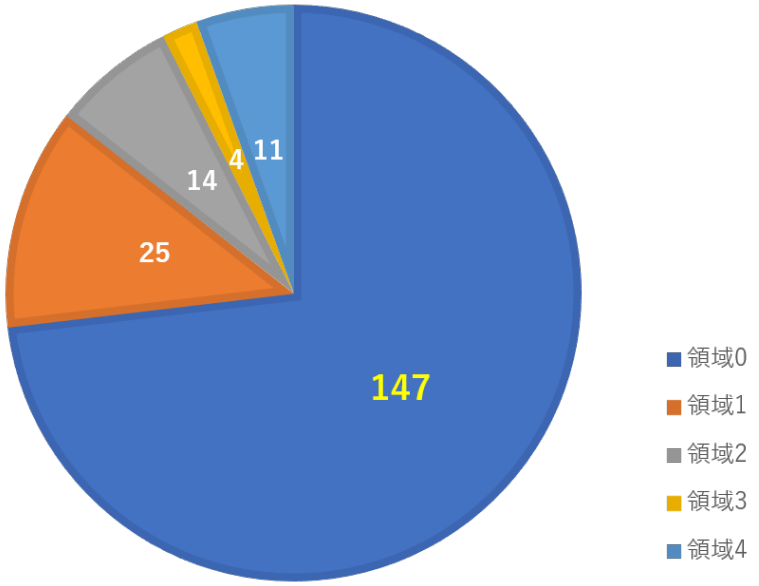


図 4.6: 誤入力の集計結果

4.3 実験 2 視覚的フィードバックの効果の検証

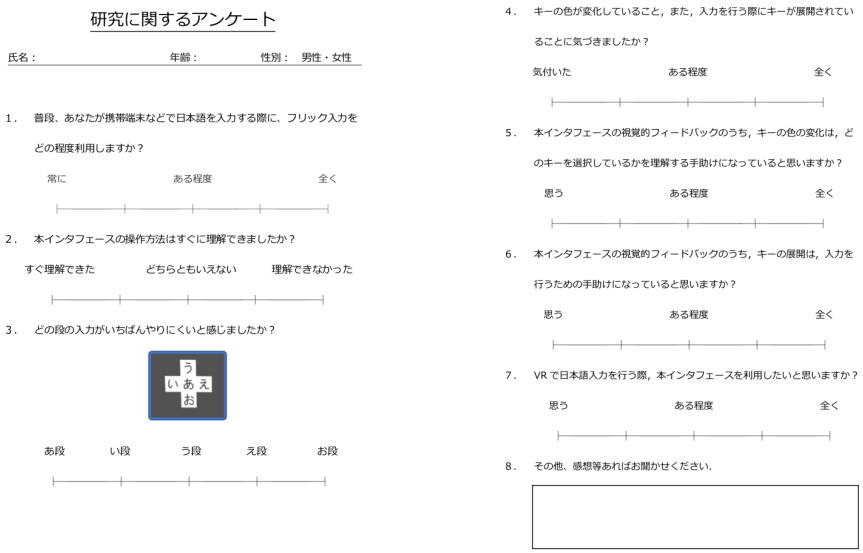
4.3.1 概要

実験 2 では視覚的フィードバックによる効果を検証する．

表 4.6 のように，本研究で作成したインタフェースにおいて，キーの押し込み表現のみを排除したインタフェース 1 と，視覚的フィードバックを全て排除したインタフェース 2 を作成し，実験 1 と同様の評価実験を行う．被験者は実験 1 の被験者のうちの都合のついた 10 名である．この 10 名を 5 名ずつ 2 つのグループに分け，グループ 1 にはインタフェース 1，2 の順に，グループ 2 には逆の順序で実験を行ってもらった．また，実験後にアンケート (図 4.7) を実施し主観評価を行う．

表 4.6: 実装している視覚的フィードバック

	キーの展開	キーの色の変化	キーの押し込み表現
インタフェース1	あり	あり	なし
インタフェース2	なし	なし	なし
実験1のインタフェース	あり	あり	あり


研究に関するアンケート

氏名：　　年齢：　　性別：　男性・女性

1．　普段、あなたが携帯端末などで日本語を入力する際に、フリック入力をどの程度利用しますか？

常に　　ある程度　　全く

2．　本インタフェースの操作方法はすぐに理解できましたか？

すぐ理解できた　　どちらともいえない　　理解できなかった

3．　どの段の入力がいちばんやりにくいと感じましたか？

う
い あ え
お

あ段　　い段　　う段　　え段　　お段

4．　キーの色が変化していること，また，入力を行う際にキーが展開されていることに気づきましたか？

気付いた　　ある程度　　全く

5．　本インタフェースの視覚的フィードバックのうち，キーの色の変化は，どのキーを選択しているかを理解する手助けになっていると思いますか？

思う　　ある程度　　全く

6．　本インタフェースの視覚的フィードバックのうち，キーの展開は，入力を行うための手助けになっていると思いますか？

思う　　ある程度　　全く

7．　VR で日本語入力を行う際，本インタフェースを利用したいと思いますか？

思う　　ある程度　　全く

8．　その他、感想等あればお聞かせください．

図 4.7: 実験アンケート

4.3.2　実験結果と考察

実験結果を表 4.7，表 4.8 に示す．また，アンケートの結果を表 4.9 にまとめた．アンケート結果について，評価が 4 以上の良いと評価されたものについては赤字で示す．また，入力速度および誤入力頻度をグラフとして視覚化した結果をそれぞれ図 4.8，図 4.9 に示す. 被験者のそれぞれについて実験 1 で行った結果を並べて示している．被験者番号については実験 1 のものと同じ番号を割り振った．入力速度については過半数以上の被験者が後に使用したインタフェースの方が素早く入力ができていることから，慣れによる入力速度の向上があると考えられ，視覚的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であった．誤入力頻度については数名の被験者について，インタフェース 1 は他の 2 つのインタフェースと比較し，誤入力の頻度が低いデータが見受けられるが統計的に考察するためには材料が少なく，正当に評価することが難しい．しかしながら，アンケートの設問 5，6 の回答を見ると，視覚的フィードバックによって入力が助けられたと感じているユーザが多く，効果があるのではないかと推測される．

あまり視覚的フィードバックによる効果が見られなかった理由として，フリック入力を普段全く利用しないと回答した被験者 7 については視覚的フィードバックによって誤入力頻度が大きく減少していることから，フリック入力が一般的な入力方法であるため，フリック入力に習熟しているユーザはほとんどインタフェー

スを見ずに入力を行っていたのではないかと推測する．

表 4.7: グループ 1 の実験結果

グループ1	インタフェース1									
	CPM	キーの入力回数	誤入力回数	誤入力率	誤入力					
					領域0	領域1	領域2	領域3	領域4	領域5
4	44	47	13	27.7%	4	6	2	0	0	1
5	47	51	6	11.8%	2	4	0	0	0	0
7	19	22	6	27.3%	3	2	0	0	0	1
15	49	55	14	25.5%	4	3	3	0	2	2
17	71	85	12	14.1%	6	5	0	0	0	1

グループ1	インタフェース2									
	CPM	キーの入力回数	誤入力回数	誤入力率	誤入力					
					領域0	領域1	領域2	領域3	領域4	領域5
4	46	52	9	17.3%	5	4	0	0	0	0
5	63	70	7	10.0%	6	0	1	0	0	0
7	24	24	17	70.8%	2	0	11	0	0	4
15	37	39	23	59.0%	13	9	0	0	1	0
17	79	89	10	11.2%	6	3	1	0	0	0

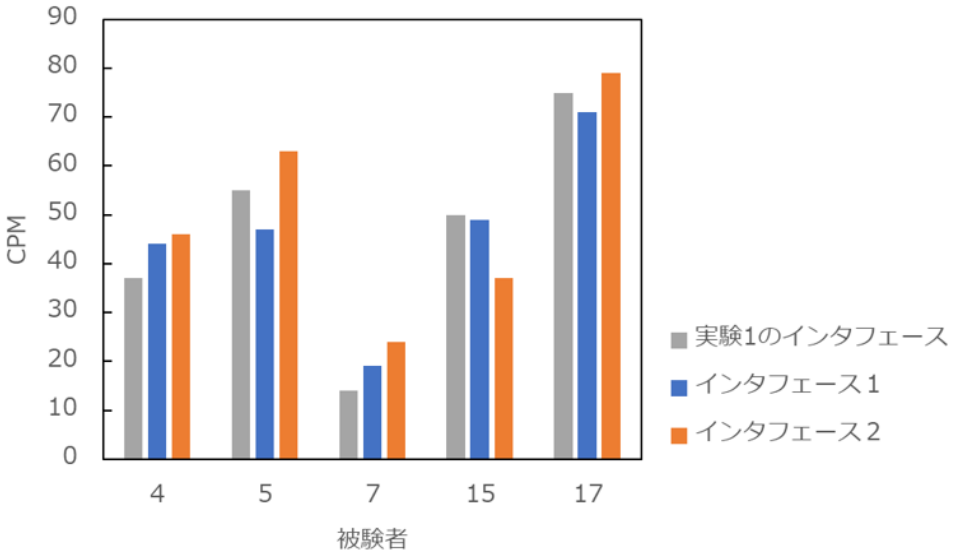
表 4.8: グループ 2 の実験結果

グループ2	インタフェース1									
	CPM	キーの入力回数	誤入力回数	誤入力率	誤入力					
					領域0	領域1	領域2	領域3	領域4	領域5
2	63	72	8	11.1%	7	1	0	0	0	0
6	52	58	9	15.5%	7	1	0	0	0	1
9	84	88	20	22.7%	17	3	0	0	0	0
11	47	53	8	15.1%	6	0	2	0	0	0
14	66	75	8	10.7%	4	1	0	0	2	1

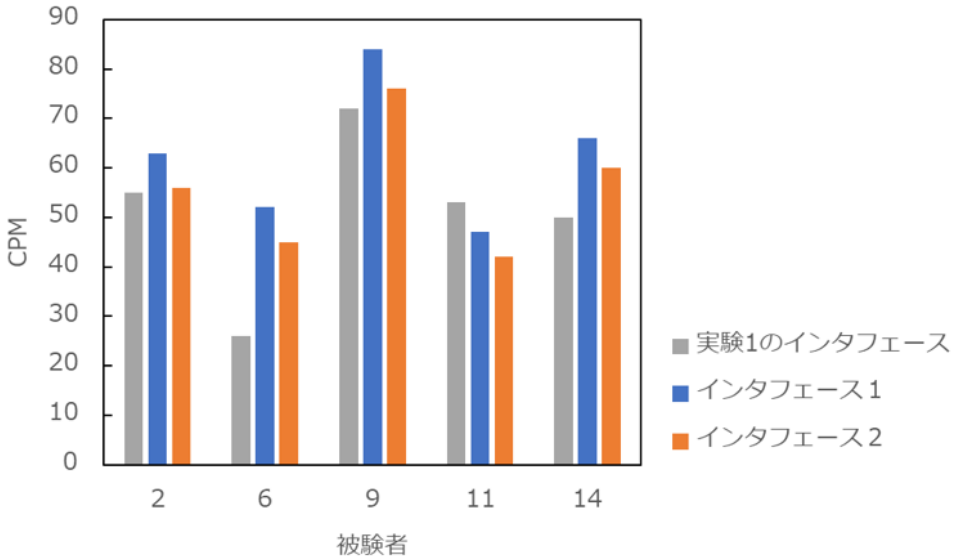
グループ2	インタフェース2									
	CPM	キーの入力回数	誤入力回数	誤入力率	誤入力					
					領域0	領域1	領域2	領域3	領域4	領域5
2	56	65	8	12.3%	5	3	0	0	0	0
6	45	50	7	14.0%	4	2	0	0	0	1
9	76	84	18	21.4%	17	1	0	0	0	0
11	42	48	10	20.8%	4	2	4	0	0	0
14	60	65	35	53.8%	16	1	1	2	15	0

表 4.9: アンケート結果

被験者		アンケート結果						
		設問1	設問2	設問3	設問4	設問5	設問6	設問7
グループ1	4	5	5	う	5	5	5	5
	5	5	5	い	5	5	5	3
	7	1	5	お	5	3	3	3
	15	5	5	あ	5	5	5	3
	17	5	5	あ	5	3	5	4
グループ2	2	5	5	あ	5	5	5	4
	6	5	5	あ	5	5	4	5
	9	5	5	お	5	5	2	4
	11	5	5	あ	5	4	5	4
	14	5	5	あ	5	5	5	4
平均値		4.6	5	ー	5	4.5	4.4	3.9

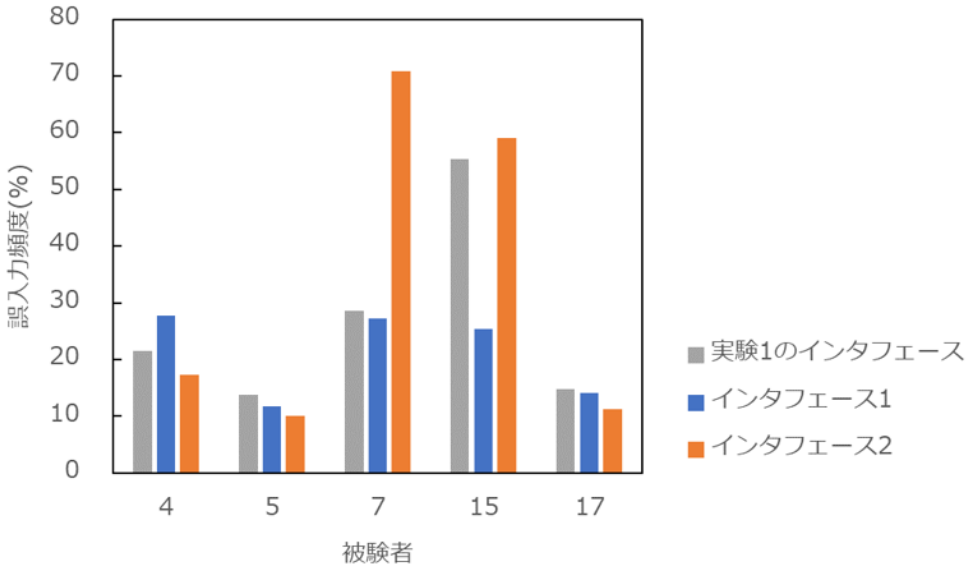


(1) グループ 1 の実験結果

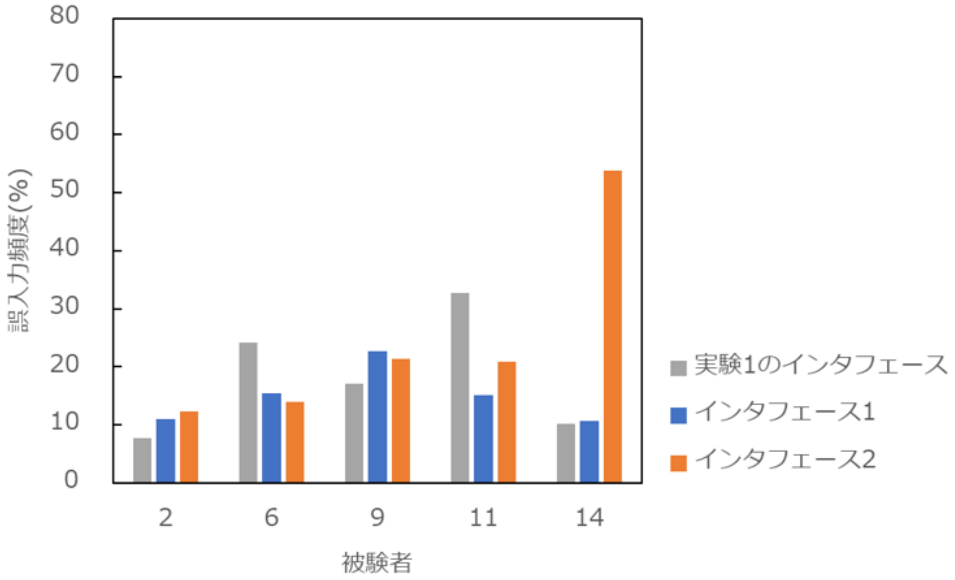


(2) グループ 2 の実験結果

図 4.8: 入力速度についてのインタフェースごとの比較



(1) グループ 1 の実験結果



(2) グループ 2 の実験結果

図 4.9: 誤入力頻度についてのインタフェースごとの比較

第5章　結論

この章では，はじめに本論文の結論を述べ，今後の展望について述べる．

5.1　まとめ

本研究では，フリーハンドの VR 環境でスマートフォンなどの携帯端末で行うフリック入力と同様の動作で日本語ひらがなを入力できるインタフェースを作成し，様々な視覚的フィードバックを付与することで操作性の良いインタフェースにすることを目標として研究を行った．本研究で付与した視覚的フィードバックは主に 3 つあり，キーの押し込みの表現，選択しているキーの色の変化，キーの展開による入力の補助である．

作成したインタフェースを利用して，VR コンテンツでよく利用されている既存の文字入力手法と比較を行った．既存の入力手法と比較して，同等以上の速度で入力が行えることが分かったが，誤入力の頻度が非常に高く改良の余地がある．

一方，本研究で作成したインタフェースから視覚的フィードバックを取り除いたインタフェースを利用して入力実験を行ったところ，得られた計測データからは視覚的フィードバックによる効果は不明であったが，同時に行ったアンケート調査より視覚的フィードバックは入力の補助や誤入力の軽減に効果があると推測される結果を得た．

本研究によって解明できた点は必ずしも多くはないが，視覚的フィードバックの効果について，若干なりとも寄与できたと考える．

5.2　展望

スマートフォンなどの携帯端末でフリック入力を行うときの把持姿勢は複数あるが [9]，本研究で作成したインタフェースはその中の人差し指を利用した操作しかできない．また，評価実験時に被験者からスマートフォンなどの携帯端末でフリック入力を利用するのと比較して，疲労するとの意見があった．これは，スマートフォンなどの携帯端末で利用しているフリック入力形式のキーボードよりも大きなものであるので，指を動かすために腕全体を動かす必要があったためではないかと考える．

今後，誤入力頻度を減らすための入力アルゴリズム模索し，また，VR 環境にお

いてフリック入力を行う際に，操作性を向上させるための正しい視覚的フィードバックを考案，検証することでインタフェースの有用性を高めたいと考えている．さらに，インタフェース全体の小型化を行うことができれば，複数の把持姿勢に対応し，ユーザの疲労度を軽減したインタフェースになると考えている．

謝辞

本研究を進めるにあたって，様々な方にご指導，ご協力を頂きました．

まず，研究の様々な場面において，適切な意見や助言でご指導してくださった，主指導の宮田一乗先生に心より深く感謝申し上げます．

やりたい分野にこだわって，なかなかテーマを決めれなかった私に助言をして下さったり，テーマを決めた後も先見の明を持って指導して下さった浦正広先生，研究について的確な意見を下さった謝浩然先生，副テーマ研究や研究計画書の執筆の際の折々で適切な意見を下さった池田心先生，学生としての心構えや考え方のご指導をして下さったり，中間発表で情報科学分野の視点から助言を下さった小谷一孔先生に深く感謝いたします．

大学のことや研究室のことを教えてくださった先輩方，学生生活の節々で食事や娯楽に誘ってくれたり，研究に行き詰ったときに一緒に考えてくれた友人や後輩たち，また，突然のご連絡にも関わらず，迅速な対応で論文を送ってくださった岩手県立大学高田研究室の喜多さん，その他，実験に協力してくれた皆様に感謝いたします．

特に，宮田一乗先生と宮田研究室の皆様には大変感謝しています。新たな環境で新しいことに挑戦する，非常に大変な学生生活でしたが，ユーモアと知識を兼ね備えた素晴らしい先生と出会えたこと，ともに苦労し挑戦する仲間に出会えたことで大変ながらも楽しく過ごせたと思います．

最後に，改めて，自分を支えてくださった全ての皆様に感謝いたします．

参考文献

[1] 西川 善司，古林 克臣，野生の男，izm，比留間 和也，VR コンテンツ開発ガイド 2017，MDN，pp．47–96 (2017)

[2] 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-若者が PC よりもスマートフォンを好んで使用する理由の-考察-，コンピュータ＆エデュケーション，VOL.43，pp.67–72 (2017)

[3] 中村 薫，Leap Motion プログラミングガイド [改訂版]，工学社，pp.8 (2015)

[4] 細野 敬太，笹倉 万里子，田辺 浩享，川上 武志．Leap Motion を用いたジェスチャによる文字入力手法の提案，人工知能学会全国大会論文集 (2014)

[5] 小澤 宗馬，梅澤 猛，大澤 範高．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，第 14 回情報科学技術フォーラム，第 3 分冊，pp.389–390 (2015)

[6] 宮田 明広，伊與田　光宏．Leap Motion を用いた文字入力手法の提案，電子情報通信学会東京支部学生研究発表会 (2016)

[7] 二本松 拓哉, 中村 喜宏. VR 環境におけるピンチ動作を用いた視覚によらない文字入力方法の提案，情報処理学会第 79 回全国大会，pp.4-309–4-310 (2017)

[8] 喜多 修太朗，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，第 181 回ヒューマンコンピュータインタラクション研究会 (2019)

[9] 永井 美波，池川 航史，志築 文太郎，高橋 伸. 日本語フリック入力を用いたモバイル端末の把持姿勢識別, 25th Workshop on Interactive Systems and Software (WISS 2017)

[10] Kosuke Komiya，Tatsuo Nakajima. A Japanese Input Method Using Leap Motion in Virtual Reality，Tenth International Conference on Mobile Computing and Ubiquitous Network (ICMU) (2017)