

# SplitFlick: コントローラのスティック操作とボタン操作を 組み合わせた VR 環境下での文字入力手法の提案

樋口 健太<sup>†</sup> 矢田 宙生<sup>††</sup> 山名 早人<sup>†††</sup>

<sup>†</sup> 早稲田大学基幹理工学部 〒 169-8555 東京都新宿区大久保 3-4-1

<sup>††</sup> 早稲田大学院基幹理工学研究科 〒 169-8555 東京都新宿区大久保 3-4-1

<sup>†††</sup> 早稲田大学理工学術院 〒 169-8555 東京都新宿区大久保 3-4-1

E-mail: <sup>†</sup> kenta213@yama.info.waseda.ac.jp, <sup>††</sup> yada\_yuki@yama.info.waseda.ac.jp,

<sup>†††</sup> yamana@yama.info.waseda.ac.jp

あらまし VR (Virtual Reality) 内での文字入力手法は、1) コントローラや手・視線のトラッキングにより VR 内の仮想のキーボードをクリックする方法、2) 入力用グローブ等の新たなデバイスを用いる手法、3) コントローラのジョイスティックやボタン等を用いる手法に分類できる。1) はユーザの肉体的疲労が大きく、2) は新たなデバイスを必要としフレキシビリティがない。一方、3) は実現が容易であることから多くの研究が行われている。本稿では 3) の手法を採用し、初めて利用する人でも高い入力速度と低エラー率を実現できるフリック入力手法「SplitFlick」を提案する。被験者実験の結果、SplitFlick は、初めて利用する被験者で 25.1CPM (Character per minutes), 入力のエラー率 TER (Total Error Rate) 6.4%を達成した。本結果は、従来のジョイスティックを使ったフリック入力手法 JoyFlick と同等である。さらに、5 日間に渡る練習による CPM, TER の変化を調査し、JoyFlick と同等の CPM 上昇と TER 低下の傾向を確認した。一方で、JoyFlick を上回る性能を達成するには至らず、エラー率を抑えながら入力速度を上げるための知見として「操作性の向上」が必要であるとの結論に達した。

キーワード VR, UI, HCI, 文字入力

## 1. はじめに

近年、安価な VR 機器の登場により VR 技術が普及し、教育[1]やオフィスワーク[2]でも VR 技術を応用する試みがある。これらの分野で VR 技術を利用する場合、VR 内での文字入力が必要不可欠となる。

VR 内での文字入力として、現実空間のキーボードを使い文字入力を行う手法が一部の VR コンテンツ<sup>12</sup>で利用できる。しかし、現実空間のキーボードを使う手法はキーボードが置かれた机の前に座った状態でしか利用できず、利用可能な場面に限られる。そのため、より広い場面で利用可能な手法の研究が進んでいる。

従来研究は、1) コントローラや手、視線のトラッキングを利用した手法[3-5]、2) 独自に開発したデバイスを使った手法[5,6]、3) コントローラに搭載されているボタンやジョイスティック、トラックパッドを使った手法[4,5,7]に分類できる。

1) は肉体的疲労が大きく、2) は新たなデバイスの開発が必要となるが、3) は、肉体的疲労が小さく、既存のコントローラ以外のデバイスを必要としない。

3) の例として、日本語入力を対象に、コントローラのジョイスティックやトラックパッドを使ったフリック入力手法の研究がある [4,7]。こうしたフリック入力手法は、ジョイスティックを特定の方向に倒すか、円形のトラックパッド上で指を特定の位置に置く、という 1 手順で入力文字の子音を 10 種類の中から指定する。しかし、この操作は子音 1 つあたりの認識範囲が小さく、ジョイスティックやトラックパッド上の指を動かした際、意図した子音と異なる子音として認識される可能性が高い。実際にジョイスティックを使う手法[4]は、1 週間練習を行った後も高いエラー率を示している。さらに、上述のフリック入力手法はジョイスティックの倒す方向や、トラックパッド上の指の位置を全ての子音と対応付けるため、現実空間で一般的に用いられている図 1.1 とは異なる子音配置をした独自のキーボードを使う。このため、使用に際して慣れが必要となり、学習コストが高い[7]。

<sup>1</sup> <https://immersed.com/>

<sup>2</sup> <https://www.meta.com/jp/work/workrooms/>

あ	か	さ
た	な	は
ま	や	ら
	わ	

図 1.1 従来のフリック入力用キーボード

そこで本稿では、エラー率が低く、短期間の訓練で習得できる手法を目指し、図 1.1 のキーボードと同様の子音配置のキーボードを採用し、VR コントローラのジョイスティックとボタン操作と組み合わせて子音を指定し、フリック入力を実現する「SplitFlick」を提案する。本論文の貢献を以下に示す。

- 現実空間でのフリック入力と同様の子音配置のキーボードを使い、VR コントローラのジョイスティックとボタン操作を組み合わせたフリック入力を行手法のパフォーマンスについて、5 日間練習を行った時の変化を含めて明らかにした。

なお、本手法を用いるためには、両方の手に持つコントローラに、それぞれ 6 方向以上認識が可能なジョイスティックまたはトラックパッドと、2 個以上のボタンが存在する必要がある。本条件は Meta Quest2<sup>3</sup>, HTC VIVE Pro/Pro Eye<sup>4</sup>, PICO4<sup>5</sup> といった代表的な VR 機器のコントローラが満たしている。

本論文の構成は以下のとおりである。まず 2 節では関連研究について述べ、3 節では提案手法の詳細を説明する。4 節では提案手法についての短期実験について説明し、5 節では長期実験について説明する。最後に 6 節で結論を述べる。

## 2. VR 上での文字入力に関する研究

VR 上の文字入力手法に関する従来研究を紹介する。

### 2.1 トラッキングを用いる手法

現在、モーショントラッキング機能を持つ VR コントローラや、手や視線のトラッキング機能を搭載した HMD (Head Mounted Display) が登場している。これらのトラッキング機能を利用して、VR 上に表示されたキーボードから入力文字を指定する手法がトラッキングを用いた手法である。具体的には、手やコントローラでキーボードに触れる手法[3,4]や、視線をポインターとして用い、入力したい文字を指定する手法[4,5]がある。現在多くの VR コンテンツで導入されている手法も、コントローラを動かしてレーザーポインタを操作し、入力したい文字にレーザーポインタを当

てることで文字入力を行っており、Controller Pointing (以下 CP) と呼ばれる[3]。

トラッキングを利用した手法の問題点は、1) 操作中に腕や眼を常に動かす必要があり肉体的疲労が大きいことと、2) 入力パフォーマンスがトラッキングの精度に大きく依存してしまうことである。

### 2.2 新たなデバイスを用いる手法

HMD の側面に取り付けるキーボード[6]や、入力用グローブ[5]といった、独自の文字入力デバイスを開発・利用する手法の研究も存在する。しかし、これらの手法は新たなデバイスが必要となり、ユーザとメーカーにとってコストがかかる。

### 2.3 既存のコントローラ (ジョイスティックやボタン、トラックパッド) を用いる手法

既存のコントローラを用いた手法として、CP のようにモーショントラッキング機能を使った手法以外に、ボタンやジョイスティック、トラックパッドを使った手法がある。これらの手法は、トラッキング精度が原因となる入力パフォーマンスの低下は発生しない。また、腕やコントローラを持ち上げて動かす必要が無く、肉体的疲労が小さい。さらに、これらの手法はコントローラのボタン配置等に依存はするものの、既存のコントローラのみで操作が行えるため実現がしやすい。

しかし、ボタンやジョイスティック、トラックパッドを使って入力を行う場合、1 手順での文字の指定が困難となる。つまり、CP に比較して入力手順が増加し、CP と比べて入力時間が長くなりやすい。

ここで、日本語の仮名はアルファベットと異なり、子音と母音を組み合わせて任意の仮名を表現できるという特徴がある。つまり、日本語入力は「子音の指定」と「母音の指定」の最短 2 手順で文字を指定することができる。本性質を利用した代表的な文字入力手法として、フリック入力がある。以下では、コントローラを用いたフリック入力手法の既存研究について述べる。

コントローラを使った VR 空間上でのフリック入力手法として、Takahashi ら[4]が提案した RoundFlick (以下 RF) がある。RF は子音 10 種類が円形に表示されたキーボードを使う。片方のコントローラのトラックパッド上の指の位置をもとに子音を指定し、もう一方のコントローラのトラックパッド上でフリック操作を行うことで、母音を指定し文字を入力する。RF は実験の結果、CP より劣る入力速度を示している。これは、キーボードのレイアウトが現実空間のフリッ

<sup>3</sup> <https://www.meta.com/jp/quest/products/quest-2/>

<sup>4</sup> <https://htcvive.jp/category/product02.html>

<sup>5</sup> <https://www.picoxr.com/jp/products/pico4>

ク入力で用いられるレイアウトと異なることや、子音 10 種類の内から 1 つの子音を 1 手順で選択する必要があり、意図しない子音の指定が起こりやすいことなどが考えられる。

RF に似た入力手法として、Yokoyama ら[7]が提案した JoyFlick (以下 JF) があり、入力デバイスとして、Nintendo Switch Pro コントローラ<sup>6</sup>のジョイスティックを利用している。JF は VR 上での利用を想定した手法ではないが、ジョイスティックを利用していることから、VR コントローラでも同様の操作が実現できると考えられる。

JF では、図 2.2 に示すレイアウトの仮想キーボードを使い、コントローラの右手のジョイスティックを 8 方向のいずれかに倒すか、どこにも倒さないことで、9 種類の子音から 1 つを指定する。その後、左のジョイスティックでフリック操作を行うことで、入力文字を指定する。実験の結果、JF は 1 週間練習を行うことで 60CPM での入力が可能と報告されている。VR 上での CP の入力速度は 35-40CPM 程度[3]であり、JF はコントローラを使う文字入力手法として一定の有効性を示している。しかし、JF は入力のエラー率 TER [8]が 10%と高い。これは、子音指定のためにジョイスティックを倒した際、隣接する別の方向として認識されたことが原因ではないかと考えられる。また、JF のキーボードは、「な」と「わ」が同じ位置に置かれている点が、従来のフリック入力のキーボードレイアウトと異なる。

つ	あ	か	さ
ち	た	な	は
て	ま	や	ら
と	わ	ら	?

図 2.2 JF のキーボード ([7]Fig.8a をトレース)

以上、RF と JF には大きく 2 点の課題が存在する。まず、これらの手法はジョイスティックやトラックパッドを使って 1 手順で子音の指定を行う。しかし、その操作中に誤った子音の指定が起こる可能性が高く、パフォーマンスの低下につながると考えられる。次に、これらの手法で使われるキーボードのレイアウトは従来のキーボードのレイアウトと異なっており、手法の学習コストが高くなると考えられる。

### 3. コントローラのスティック操作とボタン操作を組み合わせたフリック入力手法の提案

従来手法である RF や JF の問題として、1) 従来のフリック入力を使うキーボードとは異なる子音配置のキーボードを使うため、学習コストが高いこと、2) 1 手順で 10 種類 (RF) または 9 種類 (JF) の子音の中から 1 つを指定することから、意図しない子音を指定する可能性が大きいこと、がある。本論文の提案手法 SplitFlick (以下 SF) は、短期間の訓練で高速な入力を習得でき、かつ、エラー率が低くなることを目指し、1) に対して従来のキーボードと同じ子音配置のキーボードを使い、2) に対して子音指定を 2 手順で行うことで、問題の解決を目指す。

2) に対して具体的には、提案手法では、キーボードを上下に分け、「①キーボードの上下の指定」、「②子音の指定」の 2 手順で子音の指定を行う。キーボードを分割して利用することで、②の子音指定の際には 6 種類または 4 種類の子音の中から 1 つを選択すればよく、意図しない子音の指定を減らすことができる。

SF では、①②で子音指定を行った後、③母音の指定を行い、①～③の全 3 手順で文字の入力を行う。

#### 3.1 入力文字の指定

提案手法 SF では、現実空間のフリック入力と同様の子音配置のキーボードを用いる (図 3.1)。



図 3.1 提案手法で利用するキーボード

SF では、Meta Quest 2 のコントローラのジョイスティックと 2 種類のボタンを利用して文字入力を行う。図 3.2 に、Meta Quest 2 のコントローラと、操作に使う部分の名称を示す。以降、コントローラの各部分については、図 3.2 中で示した名称の前に、左右のどちらかを示すための「右」か「左」をつけることで示す。例えば、右手用コントローラのジョイスティックは、「右ジョイスティック」と表現する。

<sup>6</sup> <https://www.nintendo.com/store/products/pro-controller/>

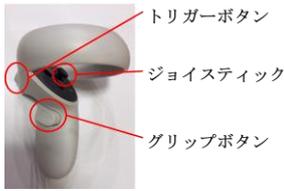


図 3.2 Meta Quest 2 のコントローラと操作に使う部分の名称

ユーザは図 3.1 に示すキーボードと図 3.2 に示された 3 種類のコントローラパーツを使い、①キーボードの上下の指定、②子音の指定、③母音の指定、という 3 手順により、キーボードから入力する文字を指定する。以下では、手順①～③について詳細に説明する。

### 手順① キーボードの上下の指定

SF では右トリガーボタンを押し、キーボードの上半分と下半分を切り替える。本手順では、ユーザは右トリガーボタンを使い、キーボードの上下どちらを使うのかを指定する。

### 手順② 子音の指定

ユーザは右ジョイスティックを特定の方向に倒すことで、入力したい文字の子音を指定する。本手順における、ジョイスティックを倒す方向と指定されるキーの位置の対応を、図 3.3 に示す。

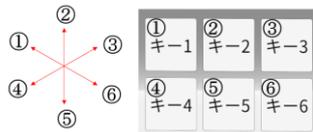


図 3.3 右ジョイスティックの倒す方向 (左) と指定されるキーの位置 (右)

### 手順③ 母音の指定

ユーザは左ジョイスティックを特定の方向に倒して母音を指定し、入力を確定する。また、母音が「あ」の文字を入力する場合には左ジョイスティックを押すことで入力ができる。本手順の、ジョイスティックを倒す方向と指定される母音の対応を図 3.4 に示す。

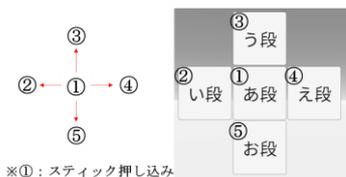


図 3.4 左ジョイスティックの倒す方向 (左) と指定されるキーの位置 (右)

以上の手順①～③により、文字の入力を行う。各手順でのキーボードの様子を図 3.5 に示す。



図 3.5 各手順でのキーボードの様子

JF や RF では、JF での「な」行と「わ」行の文字を除き、2 手順で文字の入力ができるが、SF は 1 文字の入力に 3 手順が必要であり、手順数で SF は劣る。しかし、手順①と②の操作で使うコントローラのパーツは異なっているため、SF の操作に慣れることで、手順①と②を右手でほぼ同時に行い、2 手順と同じような時間で操作ができると考える。

## 3.2 文字の削除・小文字・濁音・半濁音化

文字の削除も、コントローラのボタンを使って行うことができる仕様とした。入力操作と削除を同じコントロールパーツに割り当てた場合、誤って文字削除する場合がありますと考え、削除は両手のグリップボタンを同時に押すことで実現する。これにより、誤削除の発生を抑える。

文字の小文字・濁音・半濁音化については、変換したい文字を入力した状態で、左手のトリガーボタンを押すことで実行できる仕様とした。

## 3.3 句読点入力

SF では、仮名入力に使うキーボードに句読点キーが存在する。そのため、仮名入力と同様の手順で句読点キーを指定し、句読点を入力できる。

一方、従来手法である RF や JF は句読点入力を実装していない。RF や JF は、句読点や記号入力を行うためのキーボードを追加することで、キーボードの切り替えにより句読点入力を行う。しかし、キーボード切り替えが必要となることにより、句読点を含む文章入力では、CPM が RF や JF の論文で報告された値より小さくなると考えられる。

## 4. 短期実験

本節では提案手法である SF について、1) 従来と同等か同等以上の CPM を達成できるか、2) 従来手法と比べ低エラーを実現できるか、の 2 点について評価を行う。

### 4.1 ベースライン手法

ベースライン手法として、1) 現在広く使われている CP と、2) 従来のフリック入力手法で、エラー率が高いものの優れた入力速度を示した JF を採用した。

CPについては、50音配列を使った方式（以下CP-50）と、一般的に用いられるQWERTY配列を使った方式（以下CP-QWERTY）の2種類を実装した。

なお、SFとCPにおける比較を公平に行うために、SF2を用意した。SF2では、JFと同様に左右に分かれて表示されるキーボード（図4.1）を使い、操作方法のみSFと同様とした。これは、JFでは、母音用と子音用の2つのキーボードを左右に分けて表示しておりキーボードの表示方法を揃えるためである。



図 4.1 SF2 のキーボードレイアウト

SFとSF2、CP-QWERTYとCP-50は操作方法が同一であり、同一被験者で全ての手法の実験を行った場合、後から行った手法に慣れの影響が出る可能性が大きい。そこで本実験は、異なる被験者で実験を2回行い、1回目の実験（以下、短期実験1）ではSFとCP-QWERTY、2回目の実験（以下、短期実験2）ではSF2、JF、CP-50について、4.4の評価指標をもとにパフォーマンスを測定した。

#### 4.2 被験者

短期実験1の被験者は大学生、大学院生の男女合計16名で、年齢は18-26才（平均21.3才）、全員が右利きであった。一方、短期実験2の被験者は大学生、大学院生の男女合計13名で、年齢は19-23才（平均20.9才）、12人が右利きで1人が左利きであった。

なお、被験者の利き手によらず、同様のインタフェースで実験を行った。

#### 4.3 実験方法

本実験における被験者の実験の手順を次に示す。

- ① 被験者は、入力手法や実験内容の説明を受ける。
- ② 被験者は、1つの手法の練習を10分間行う。練習中は本番と同様にフレーズを提示したが、被験者が練習したい操作を行うことを許可した。
- ③ 被験者は、練習した手法を使い、提示されたフレーズの入力を5分間行う。フレーズは15種類用意され、10分以内に全てのフレーズの入力が終了した際は、次の手順に移動した。
- ④ 被験者は、他の手法で②、③を行う。
- ⑤ 被験者はアンケートに回答する。

最後のアンケートは、手法に対する感想を書くための自由記述欄を設けた。

疲労や慣れが原因となり、手法の利用順序が結果に影響を与える可能性がある。そこで、2つの手法を使う順番は2通り、3つの手法を使う順番は6通りあるため、短期実験1は2つのグループ、短期実験2では6つのグループに被験者を分けて実験を実施した。

フレーズは下岡ら[9]が考案したものを使い、本番では練習で提示していないフレーズを用いた。フレーズは6~8文字のひらがな（「けんきゅうしつ」、「どうねんだい」等）で構成される。

#### 4.4 評価指標

各手法の入力パフォーマンスを調べるため、1分当たりの入力文字数であるCPMと、入力のエラー率TERを測定した。CPMは以下の式(1)で計算される。

$$CPM = \frac{\text{提示フレーズの文字数}}{\text{入力にかかった時間[分]}} \quad (1)$$

なお、入力にかかった時間は入力を始めてから全ての文字が入力されるまでの時間を自動で計測した。

本実験では、提示したフレーズと完全に一致するように入力を行ったため、全ての誤入力は修正された。このとき、TERは、以下の式(2)で計算される。

$$TER = \frac{\text{削除回数}}{\text{入力した文字数}} \times 100 \quad (2)$$

本実験では、例えば「どうねんだい」と入力する際に誤って「どうねんない」と入力した場合、被験者は2文字削除し「どうねん」とした後に入力を続行した。ただし、上の例では「どうねんな」と入力した時点でブザー音になる仕様とし、誤入力をしたときには即座に被験者が気づけるようにした。

#### 4.5 結果と考察

各手法の結果を図4.2、図4.3、表4.1に示す。また、有意水準を0.05としてボンフェローニ法を使い、各手法間で有意差の有無を求めた。有意差が確認できた手法の組み合わせを図中に線で見ないで示す。

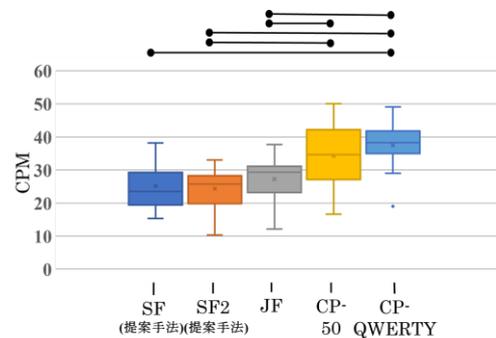


図 4.2 各手法の CPM

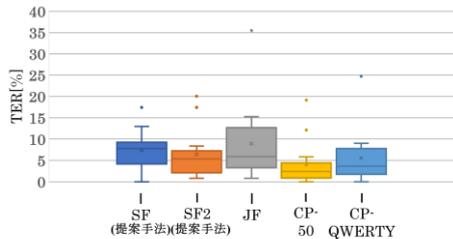


図 4.3 各手法の TER

表 4.1 各手法の平均 CPM と平均 TER

手法	CPM	TER [%]
SF (提案手法)	25.1	6.4
SF2 (提案手法)	24.4	6.4
JF	27.3	8.9
CP-50	34.2	4.0
CP-QWERTY	37.5	5.5

実験の結果を次にまとめる。

#### 1) SF, SF2 と JF は入力速度とエラー率が同等

SF と SF2, JF 間では CPM・TER とともに有意差が認められない。これは、SF, SF2, JF は操作方法が似ており、被験者がどの手法にも慣れることができなかつたためだと考えられる。

#### 2) 提案手法 (SF, SF2) は CP (CP-50, CP-QWERTY) と比べて入力速度が遅い

提案手法と CP の結果を比較すると、TER は同等であるものの、CPM は CP の方が有意に大きい。これは、CP と比べ提案手法は操作が複雑であり、被験者が提案手法の操作に慣れることができなかつたためだと考える。また、アンケートにて、「母音が「あ」の文字を指定する際のジョイスティックを押し込む操作がやりづらい」という意見があった。実験中も、ジョイスティックを押し込む際にジョイスティックが倒れて誤入力が発生し、その修正に時間を割かれている様子が見られた。そのため、CP と比べた SF のパフォーマンスの低さについて、ジョイスティックの押しこみ操作が関係している可能性がある。

本実験は、提案手法が 1) 従来と同等か同等以上の CPM を達成できるか、2) 従来手法と比べ低エラーを実現できるか、を確認するための実験であった。この目的に対し、実験結果から次のことが分かる。

- 1) について、提案手法の SF や SF2 は従来のフリック手法 JF と CPM が同等なものの、現在利用されている手法である CP よりも CPM が有意に小さい。
- 2) について、提案手法の SF や SF2 は従来手法とエラー率が同等であり、提案手法のエラー率が小さくなっていることは確認できなかった。

## 5. 長期実験

本節では提案手法である SF で目指した、短期間での手法習得が可能かどうか、を評価するために実施した長期実験について説明する。本実験では、SF とベースライン手法を被験者に 5 日間練習してもらい、各手法の CPM と TER の変化を調査した。

### 5.1 ベースライン手法

ベースライン手法は SF2, JF, CP-50 を採用した。CP-QWERTY は、Hutama らの実験結果[6]から、CPM がほぼ向上しないことが分かっているため、本実験の対象外とした。一方で、ユーザは QWERTY 配列に比べ 50 音配列に慣れていない可能性がある。そのため、CP-50 については入力スピードが向上する可能性があるとして、ベースライン手法の 1 つとした。

本実験は短期実験と同様に、操作方法が同じである SF と SF2 を同じ被験者が実施することを防ぐために 2 回に分けて行った。1 回目の実験（以下、長期実験 1）では SF, 2 回目の実験（以下、長期実験 2）では SF2, JF, CP-50 の CPM と TER を測定した。

### 5.2 被験者

長期実験 1 の被験者は大学生、大学院生 2 名で、年齢は 22, 23 才、2 人とも右利きであった。一方、長期実験 2 の被験者は大学生 4 名、年齢は 20-25 才（平均 22 才）、3 人が右利きで 1 人が左利きであった。

長期実験 1, 2 の全ての被験者がスマートフォンでのフリック入力を 1 年以上頻繁に利用していた。

短期実験と同様に、被験者の利き手によらず、同様のインタフェースで実験を行った。

### 5.3 実験方法

本実験は、各被験者に対して次の手順により、連続して 2 日以上の間隔を開けず、5 日間実施した。

#### 1 日目

- ① 被験者は、入力手法や実験内容の説明を受ける。
- ② 被験者は 1 つの手法の練習を 10 分間行う。練習中は本番と同様にフレーズを提示したが、被験者が練習したい操作を行うことを許可した。
- ③ 被験者は、練習した手法を使って、提示されたフレーズの入力を 10 分間行う。フレーズは 50 種類用意され、10 分以内に全てのフレーズの入力が終了した際は、次の手順に移動した。
- ④ 長期実験 2 の被験者は他の手法で②, ③を行う。

#### 2-4 日目

- ① 被験者は、1 日目の②-④と同じ時間で練習とフレーズの入力を行う。

#### 5 日目

- ① 被験者は各手法について、5 分間の練習の後、10 分間のフレーズ入力を行う。

② 被験者はアンケートに回答する。  
 フレーズセットは短期実験と同様のものを用いた。

### 5.4 結果と考察

本実験の、実験日数に対する各手法の平均 CPM の変化を表 5.1 に、平均 TER の変化を表 5.2 に示す。

表 5.1 各手法の平均 CPM の変化

	1 日目	2 日目	3 日目	4 日目	5 日目
SF (提案 手法)	21.5	32.0	37.6	38.3	45.0
SF2 (提案 手法)	20.8	31.1	35.5	43.9	47.2
JF	26.4	37.4	42.7	51.6	54.2
CP-50	37.3	42.1	45.1	52.7	54.1

表 5.2 各手法の平均 TER の変化

	1 日目	2 日目	3 日目	4 日目	5 日目
SF (提案 手法)	2.5	2.4	5.0	4.6	2.8
SF2 (提案 手法)	9.7	9.1	8.3	7.7	6.3
JF	10.4	8.8	8.9	6.1	6.4
CP-50	2.9	4.1	4.1	1.9	1.8

表 5.1, 5.2 は長期実験 1 の参加者 2 名, 長期実験 2 の参加者 4 名の平均値を示す。しかし、人数の少なから平均値での議論ができないため、各被験者の結果を図 5.1, 図 5.2, 表 5.3, 表 5.4 に示し、比較する。ここで、被験者を揃えるため、異なる被験者で実験を行った SF を除外して比較する。つまり、以降は長期実験 2 で実施した手法の結果のみを用いて比較する。なお、表及び図中の被験者 A が左利きで、それ以外が右利きであった。

表 5.3 長期実験 2 における各被験者の CPM

(a)被験者Aの結果						(b)被験者Bの結果					
手法	1日目	2日目	3日目	4日目	5日目	手法	1日目	2日目	3日目	4日目	5日目
SF2 (提案手法)	12.2	20.2	18.0	28.0	20.01	SF2 (提案手法)	26.8	38.7	50.1	55.8	64.4
JF	15.6	26.0	20.7	34.8	35.4	JF	34.2	46.3	56.3	70.7	69.3
CP-50	15.9	18.5	17.9	30.9	28.3	CP-50	46.6	54.7	63.2	65.9	65.7

(c)被験者Cの結果						(d)被験者Dの結果					
手法	1日目	2日目	3日目	4日目	5日目	手法	1日目	2日目	3日目	4日目	5日目
SF2 (提案手法)	18.9	32.3	35.3	44.4	51.5	SF2 (提案手法)	25.5	33.1	38.5	47.3	52.8
JF	26.5	37.9	47.4	49.3	60.2	JF	29.3	39.4	46.4	51.7	52.0
CP-50	47.5	49.4	53.2	61.5	62.2	CP-50	39.1	45.7	46.0	52.7	60.2

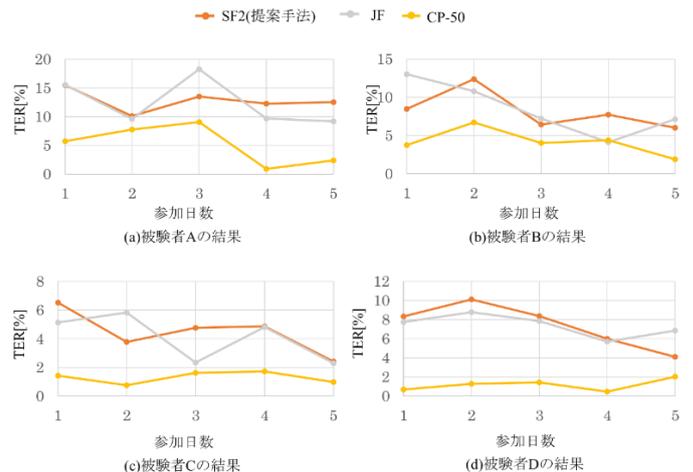


図 5.2 長期実験 2 における各被験者の TER の変化

表 5.4 長期実験 2 における各被験者の TER

(a)被験者Aの結果						(b)被験者Bの結果					
手法	1日目	2日目	3日目	4日目	5日目	手法	1日目	2日目	3日目	4日目	5日目
SF2 (提案手法)	15.5	10.2	13.5	12.3	12.5	SF2 (提案手法)	8.5	12.4	6.5	7.7	6.0
JF	15.6	9.6	18.3	9.7	9.2	JF	13.0	10.8	7.2	4.1	7.2
CP-50	5.7	7.8	9.1	0.9	2.4	CP-50	3.8	6.7	4.0	4.4	1.9

(c)被験者Cの結果						(d)被験者Dの結果					
手法	1日目	2日目	3日目	4日目	5日目	手法	1日目	2日目	3日目	4日目	5日目
SF2 (提案手法)	6.5	3.8	4.8	4.9	2.4	SF2 (提案手法)	8.4	10.1	8.4	6.0	4.1
JF	5.1	5.8	2.3	4.8	2.3	JF	7.7	8.8	7.9	5.7	6.9
CP-50	1.4	0.8	1.6	1.7	1.0	CP-50	0.7	1.3	1.4	0.5	2.0

実験結果を次にまとめる。

#### 1) SF2 と JF で、CPM の増加量は似た傾向がある

図 5.1 と表 5.3 から、被験者 A 以外、つまり右利きの被験者 3 人は SF2 と JF を使った時の CPM が同程度の傾きで増加している。提案手法は JF よりも小さい学習コストの達成を目的としたが、本目的の達成を確認することはできなかった。なお、左利きの被験者 A は、最後の 5 日目で CPM が下がっており、また利き手が他の 3 名と異なることから、同結果を持って議論することが困難な結果となった。

本結果については、提案手法が採用した「キーボードのレイアウト変更による学習コストの低下」を、「提案手法のキーボード切替操作による学習コストの

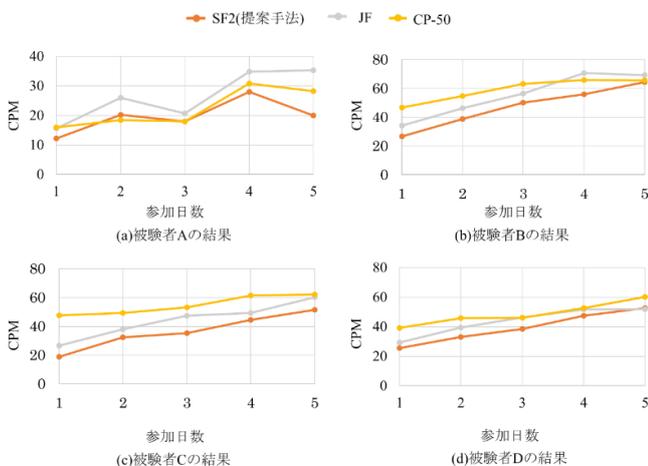


図 5.1 長期実験 2 における各被験者の CPM の変化

## 参 考 文 献

増加」が打ち消したことが原因ではないかと考えられる。1手順で子音指定を行う JF に対し、SF2 は子音指定時に①キーボードの上下の指定、②子音の指定の 2 手順が必要である。しかし、4・5 日目の実験でも、被験者が提案手法利用時に、誤ってキーボードを切り替えてしまうことがあった。そのため、「適切にキーボードを切り替える」という操作の学習コストが大きく、当初の目的の達成を確認できなかったと考えられる。

### 2) JF と SF2 の間で、TER 減少量は似た傾向がある

図 5.2 と表 5.4 から、日ごとに SF2 と JF の TER の大小関係が変化しており、TER の変化量が日により大きく異なる。左利きの被験者 A を除き、被験者 B, C, D に限って TER の変化を見た場合も、SF2 と JF2 の間で差を確認することは困難であった。以上から、SF2 と JF では、学習日数に対する TER 減少が似た傾向であることは確認できるが、差については確認できなかった。

以上から、提案手法は、JF と比べた時の学習効果の向上を目指したが、今回の長期実験では確認することができなかったことが分かる。

## 6. おわりに

提案手法は従来手法の問題であった、1) 従来のフリック入力と違う子音配置のキーボードを使うことによる学習コストの増加と、2) 子音指定を 1 手順で行うことによる高いエラー率とを改善するため、1) に対して従来のフリック入力と同様のキーボードを使い、2) に対して子音指定を 2 手順で行うこと、を適用した。しかし 1) は、子音指定を 2 手順で行うことによる学習コストの増加効果により、提案手法自体の学習コストが高くなり、キーボード変更の効果が確認できなかった。2) は、本実験の範囲ではエラー率改善の効果が確認できなかった。

1) の結果から、学習コスト低減にはキーボードのレイアウト以上に、操作性が関係していると考えられる。操作性の影響について正確に考察するため、今後は提案手法の各操作で、どの程度の時間がかかっているかを調査する必要があると考える。また、2) の結果から、子音指定のエラーを改善するためには、単にすべての子音の認識範囲を広げるのではなく、特定の子音指定にエラーが偏っている可能性を考慮し、それに合わせた認識範囲・操作性の調整を行う必要があるのではないかと考える。

本稿の長期実験の結果については統計的な検定が実施できるほどのデータを収集できなかったため、今後の更なる被験者実験により、得られた結果が統計的に有意かどうかを検証する必要がある。

- [1] C. Ma, Y. Du, D. Teng, J. Chen, H. Wang and D. Guozhong, "An adaptive sketching user interface for education system in virtual reality," Proceedings of 2009 IEEE International Symposium on IT in Medicine & Education, pp. 796-802, 2009.
- [2] J. Grubert, E. Ofek, M. Pahud and P. O. Kristensson, "The Office of the Future: Virtual, Portable, and Global," IEEE Computer Graphics and Applications, vol. 38, no. 6, pp. 125-133, 2018.
- [3] M. Speicher, A. M. Feit, P. Ziegler, and A. Krüger, "Selection-based Text Entry in Virtual Reality," Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, no. 647, pp. 1-13, 2018.
- [4] R. Takahashi, S. Shirai, J. Orlosky, Y. Uranishi, and H. Takemura, "A Japanese Character Flick-Input Interface for Entering Text in VR," Proceedings of 2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct, pp. 251-253, 2021.
- [5] T. J. Dube, and A. S. Arif, "Text Entry in Virtual Reality: A Comprehensive Review of the Literature," Human-Computer Interaction. Recognition and Interaction Technologies. HCII 2019, Lecture Notes in Computer Science, vol. 11567, pp. 419-437, 2019.
- [6] W. Hutama, H. Harashima, H. Ishikawa, and H. Manabe, "HMK: Head-Mounted-Keyboard for Text Input in Virtual or Augmented Reality," Proceedings of the Adjunct Publication of the 34th Annual ACM Symposium on User Interface Software and Technology, pp. 115-117, 2021.
- [7] K. Yokoyama, R. Takakura, and B. Shizuki, "JoyFlick: Japanese Text Entry Using Dual Joysticks for Flick Input Users," Human-Computer Interaction - INTERACT 2021. Lecture Notes in Computer Science, vol. 12934, pp.1-19, 2021.
- [8] A. S. Arif, and W. Stuerzlinger, "Analysis of text entry performance metrics," Proceedings of 2009 IEEE Toronto International Conference Science and Technology for Humanity, pp. 100-105, 2009.
- [9] 下岡純也, 山名早人, "ShuttleBoard:スマートウォッチにおけるタップ動作の少ない仮名文字入力手法," 日本データベース学会和文論文誌, vol.16-J, no. 5, pp. 1-7, 2018.