

## 両手の指と掌の接触を用いたかな文字入力手法の検討

井上 賢人<sup>†</sup> 梅澤 猛<sup>‡</sup> 大澤 範高<sup>‡</sup>千葉大学工学部情報画像学科<sup>†</sup> 千葉大学大学院工学研究院<sup>‡</sup>

## 1 はじめに

非透過型 HMD(Head Mounted Display)を使用しながら文字を入力する手段として、ジェスチャ動作による文字入力手法が求められている。細野らは Leap Motion によって取得した指の移動角度を使ってかな文字の入力を行う手法を提案している[1]。また小澤らは空中でのジェスチャによりフリック入力を行う手法を提案している[2]。フリック時のキー選択をつまむ動作で行うことで、画面タップに代わる手応えを指同士の接触によって得られると述べられている。

本研究では指の接触に着目し、両手の指と掌の接触を用いて、五十音の中から歴史的仮名遣いを除いた 46 文字を入力する手法について検討する。

## 2 親指の接触を利用した文字入力

親指でそれ以外の 4 指の指先および掌に触れることで、対応する箇所を設定した要素を選択する。1 文字の入力は、(1) 子音選択、(2) 母音選択、(3) 文字決定の 3 つのステップで行い、触れる位置に設定される要素は各ステップで異なるものとする。

子音選択ステップでは、右手の人差し指から順に「あ行」から「た行」、掌に「な行」を割り当てる。左手も同様に「は行」から「わ行」を割り当てる。(図 1)

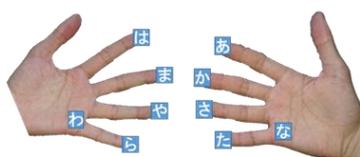


図 1 子音選択ステップ

左右いずれかの親指で、入力したい文字の子音を選択することで母音選択ステップ(図 2)へと移行する。

母音選択ステップでは、子音選択で使用していない方の手の人差し指から順に「あ段」から

Study on input method for kana characters by touching thumb to fingers with both hands

<sup>†</sup> Kento Inoue, Department of Informatics and Imaging Systems, Faculty of Engineering, Chiba University

<sup>‡</sup> Takeshi Umezawa, Noritaka Osawa, Graduate School of Engineering, Chiba University

「え段」、掌に「お段」の母音を割り当てる。



図 2 母音選択ステップ(「な行」の場合)

入力したい文字の母音を選択することで文字決定ステップへと移行する。

文字決定ステップ(図 3)では、両手の親指を離すことで決定した文字が入力され、再び子音選択ステップへと移行する。



図 3 文字決定ステップ(「に」の場合)

これらの 3 つのステップを繰り返すことで文章の入力を行う。なお、入力した文字の削除は「わ行」の「お段」に割り当てた。

## 3 要素の選択判定

前述の文字入力手法を使った文章入力アプリケーションを Unity 上に実装した。指の動作検出には Leap Motion を使い、検出頻度は Unity のレンダリングフレーム(以下フレーム)ごととした。フレーム長は動作環境によって変動し、今回の環境では平均 20ms であった。Leap Motion で検出された親指の先が、それ以外の 4 指の指先および小指の付け根のいずれかと一定の距離以内となったときを接触とした。Leap Motion に対する手の傾きによっては指の認識精度が低下し、実際には中指に触れていても薬指への接触と誤認識するフレームが確認された。そのため、連続する複数のフレームを使って判定を行う方法 2 種類を比較検討した。

## 3.1 フレーム数による判定

複数のフレームを 1 つのブロックとし、ブロック内で最も多く推定がされたものを親指と接触したと判定をする。ブロックは図 4 のように移動ウィンドウ方式でフレームごとに判定を行う。

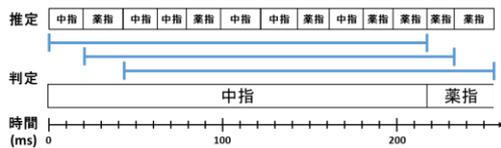


図4 フレーム数による判定

### 3.2 占有時間による判定

フレームごとの判定ではフレーム長が動作環境によって変動するため、判定に要する時間も変わってしまう。そこで、判定に要する時間を安定化させるため、単位時間内で推定されるフレームの合計時間が一番長いものを接触対象と判定する。ウィンドウは図5のように、Unityのフレームレートに依存しない間隔で移動し判定を行う。間隔は、デフォルト値 20ms とする。



図5 占有時間による判定

## 4 操作実験

### 4.1 参加者・実験条件

前述の2つの判定方式に加え比較対象としてゲームコントローラを用いた五十音表を使った入力によって、文字入力を行う実験を行った。実験参加者は22~24歳までの男性3名であり、全員が右利きであった。今回フレーム数による判定では11回のフレームを1つのブロックとし、占有時間による判定の単位時間を220msとした。

### 4.2 実験内容

「あ」の一文字を入力することを測定開始動作とし、続けて「おはようこんにちは」の9文字を正しく入力するまでを入力時間として測定した。入力中に文字の削除を行った回数を誤入力回数として記録した。

入力に用いる手法は、ゲームコントローラ、フレーム数による判定、占有時間による判定の順とし、対象となる手法を使って10分間の自由文による入力練習を行った後、それぞれの手法で3回ずつの測定を行った。

### 4.3 実験結果

各参加者の平均入力速度 (CPM: Characters Per Minute) は、図6の通りであった。また、全参加者の平均誤入力回数はゲームコントローラを用いた入力で0.1回、フレーム数による判定を用いた入力で1.1回、占有時間による判定を用いた入力で1.3回であった。

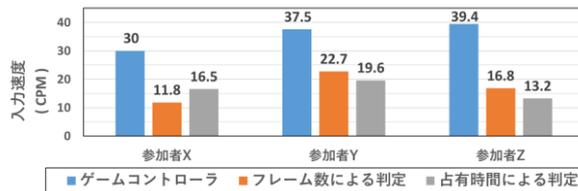


図6 参加者の各手法での平均入力速度

## 4.4 考察

ゲームコントローラを用いた入力に比べ、フレーム数による判定を用いた入力と占有時間による判定を用いた入力は共に誤入力回数が多くなった。誤入力は「に」、「ち」の入力の際に判定が「ぬ」、「つ」に切り替わってしまうことで発生していた。11回の推定の内訳を調べたところ、中指と薬指それぞれの推定数が拮抗していた。そのために判定が安定しなかったことが分かった。「に」、「ち」は親指と中指を接触させる必要があり、参加者によっては接触させる際に薬指を曲げてしまうことがあった。そのため、親指と薬指との距離が近くなり、親指と薬指が接触していると推定される数が増え判定が安定しなかったと考えられる。

フレーム数による判定と占有時間による判定の入力速度の標準偏差を求めるとフレーム数による判定では5.4CPM、占有時間による判定では3.4CPMとなり、判定速度が固定化されている方が入力速度の散らばりは小さいことが分かった。

## 5. おわりに

本稿では、両手の指と掌の接触を用いた入力手法をUnity上で実装した。判定を安定させるため、連続する複数のフレームを使って判定を行う方法2種類を実装し操作実験を行った。実験では各入力手法での入力速度と誤入力回数を測定し比較を行った。実験結果より親指の先が、それ以外の4指の指先および小指の付け根のいずれかと一定の距離以内かどうかで接触の推定を行うことは、入力する文字によっては判定の安定化に寄与しないことが分かった。

今後、接触の推定に各指の関節位置情報も利用することで推定精度の向上を図る予定である。

### 参考文献

- [1] 細野 敬太, 笹倉 万里子, 田邊 浩亨, 川上 武志, “Leap Motion を用いたジェスチャ操作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 28 巻, pp1-4, 2014
- [2] 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”, 情報科学技術フォーラム講演論文集 14 巻 3 号, pp389-390, 2015