

平成28年度 卒業論文

LeapMotionを用いた文字入力インタフェースにおける
文字選択手法

平成29年2月21日

13111002

安部 弘樹

指導教員 三浦 元喜 准教授

九州工業大学 工学部 総合システム工学科

概要

本研究では、空中タイピングにおける視覚的フィードバックの影響とその際のタイピング方式について開発と比較を行った。既存の研究では、空中タイピングは、視覚的フィードバックが少なくタイピング技術が優れている人向けに、設計されている。今回の研究では、タイピングが不慣れな人でも使いやすいものを設計することを目的として開発していった。

提案手法として、2つ考案した。一つ目は10本指キータップ手法、2つ目は、一本指タイピング手法である。どちらの手法も実際の物理キーボードの入力方式から考案した。2つ手法設計した理由は、空中での文字入力には触覚フィードバックがないことにより、10本指でキータップして入力することが最善とは言えないからである。LeapMotionを用いて設計した。

短い単語を対象とした実験の結果、入力速度と精度は、10本指キータップ手法では平均5.53WPMと80.74%、一本指タイピング方式では、7.40WPMと93.44%であった。一本指タイピング手法の方が、誰にでも練習なしで入力しやすいという結果になった。10本指キータップ手法は、練習すれば速度は上がるという結果が得られた。10本指キータップ手法は、何度も使用する機会がある人にとっては有効な手段であるという結果になった。

目次

| | |
|------------------------------|-----------|
| 第1章 序論 | 3 |
| 1.1 背景 | 3 |
| 1.2 空中タイピングについて | 4 |
| 1.2.1 有用な場面 | 4 |
| 1.2.2 空中タイピングの短所 | 4 |
| 1.3 本研究の目的 | 5 |
| 第2章 関連研究 | 6 |
| 2.1 空中仮想キーボード AirKey | 6 |
| 2.2 ATK:空中仮想キーボード | 7 |
| 2.3 携帯端末向け多指 AR タイピングインタフェース | 8 |
| 第3章 設計準備 | 10 |
| 3.1 LeapMotion | 10 |
| 3.1.1 手,指を検出 | 11 |
| 3.1.2 ジェスチャーの検出 | 12 |
| 3.1.3 座標 | 13 |
| 3.2 キーの配置と入力方法 | 14 |
| 3.3 開発環境 | 15 |
| 第4章 提案手法の設計 | 16 |

| | | |
|--------------|-------------------------|-----------|
| 4.1 | 提案手法 1 (10 本指キータップ入力手法) | 16 |
| 4.1.1 | 10 本の指の位置を常にとる | 17 |
| 4.1.2 | キーボード、指の位置の表示 | 17 |
| 4.1.3 | どの指でキータップが発生したかを判定 | 20 |
| 4.1.4 | どのキートップを打鍵したかを判別 | 24 |
| 4.1.5 | 文字を出力 | 24 |
| 4.2 | 提案手法 2 (一本指タイピング手法) | 26 |
| 4.2.1 | ターゲットの指を判別 | 28 |
| 4.2.2 | 手の動きにより打鍵したかを判別 | 28 |
| 第 5 章 | 評価実験 | 30 |
| 5.1 | 実験目的 | 30 |
| 5.2 | 実験方法 | 31 |
| 5.3 | 実験結果 | 32 |
| 5.4 | 考察 | 36 |
| 第 6 章 | 結論 | 39 |
| | 謝辞 | 41 |
| | 参考文献 | 42 |

第1章 序論

本論文では視覚的フィードバックを含む、空中タイピングシステムの提案と比較を行っていく。この章では、研究の背景、空中タイピングについて、本研究の目的について述べる。

1.1 背景

近年、PlayStationVR[1]の発売とともに、仮想現実というものは注目を浴びつつある。仮想現実は今はまだ、ゲームや動画などの現実感の生成においてにしか普及しておらず、話題になっていない。しかし、今後仮想現実の用途は多くなると予想される。例えば、医療の世界で社会不安や飛行恐怖症、会話恐怖症、高所恐怖症などを持つ患者の認知行動療法に仮想現実を利用するという事例もある。その他にも、教育やエンターテインメントなどの様々な分野で用途が増えていくだろう。

その中で仮想現実上を使用した文字入力動作に関してはまだ、あまり研究されていない。既存の研究では、フリーハンドではなく、手に器具をつけたり、手に持って文字入力を行う研究が多い。問題点として、機器を身に付ける煩わしさ、特殊な入力方式による操作訓練の必要性があげられる。この問題は、いつでもだれでも、使いやすいものではなく、シーンや使う人が絞り込まれている。私は近い未来に仮想現実が普及していくと考え、いろいろな人に使ってもらえるように空中フリーハンドタイピングの研究を行おうと考えた。

1.2 空中タイピングについて

空中タイピングは仮想現実上のみではなく実際の生活の中でも有用である。しかし、現時点ではあまり使用されていない。この節では、どのような場面で有用であるか、空中タイピングの長所、短所について述べる

1.2.1 有用な場面

空中タイピングは以下のような場面で有用である。

- VR など仮想現実上で文字を入力する場合
- コストの面で大型ディスプレイ導入時にタッチパネル機能を導入できない場合
- 小型化が進む携帯電話などポストディスプレイでの文字が入力しにくい場合
- 実際の物理キーボードが使いづらい場面。
机にキーボードを置く場所がない場合。
教科書などを見ながらメモをしたい場合。
手が汚れている場合。

上記のように多くの場面で有用である。

1.2.2 空中タイピングの短所

空中タイピングの短所として、以下の3つが挙げられる。

- 特殊な入力方式。
入力方式が一般的なキータップや押し込む、フリックではないために練習がいる。

- 手の装着感。

ハンドグローブなどをつけて手の解析を行う。

手が汚れているなど日常的にすぐには使いにくい。

- 使いにくさ。

フィードバックが少ない。

センサーを持ち運べない。

上記の短所の中でもフィードバックについてはあまり研究されていない。物理キーボードでは、触覚的フィードバック（キーを触る）と視覚的フィードバック（キーを目で追っている）、聴覚的フィードバック（押し込んだ時の音）の3つのフィードバックにより、キー入力を補助している。しかし、空中でタイピングを行うことは、仮想の空間で行っているため、3つのフィードバックは少なくなっている。

1.3 本研究の目的

空中タイピングは、フィードバックが少ないことが問題点である。フィードバックを多く含むことで、タイピングに慣れていない人にとっても使いやすいのではないかと考え、聴覚的フィードバック、視覚的フィードバックの2つを含むことを考えていく。また、特殊なタイピング方式は操作訓練の必要性がある為、慣れているタイピング方式で設計していこうと考えている。

本研究では、空中タイピングにおける視覚的フィードバックの影響とその際のタイピング方式について開発と比較を行う。

第2章 関連研究

手にデバイスを装着しない、いわゆるフリーハンドの状況で空中タイピングを行う技術についての研究はいくつかされている。本章では、それらの空中タイピング技術の関連研究について述べる。

2.1 空中仮想キーボード AirKey

星野 [2] は, 空中仮想キーボード AirKey を開発した。LeapMotion を用いて手の動きを取り、キーボードをタイプするジェスチャによるキー入力を可能にした。AirKey のシステムの構成は、まず指の座標を取得しホームポジションを設定する。次にホームポジションとの座標の差をとることで、どの指でキータップが発生したかを判定する。最後に、その指の座標がホームポジションと比較しどのキーを入力したかを判定、打鍵している。空中でタイピングしてもらうのでフィードバックの少なさを補うために音でフィードバックをしている。また、キー入力の認識率を向上するためには、それぞれの指で打鍵できるキーを指定してある。それぞれの指でのキーの指定方法は、キーボードのホームポジションと一緒である。

AirKey により

- ・手が汚れている
- ・机の上が狭い場所
- ・ノートを見ながら文字を入力したいときなど

での文字入力を可能にした。実験結果より、最初の一回目では時間がかかってしまった

が、練習をすれば7文字以内の入力であればキーボードと同様のスピードで入力できることを示した。

2.2 ATK:空中仮想キーボード

Yiら [3] は、ATK（空中仮想キーボード）の設計のなかで、空中フリーハンドタイピングを可能にする新たなキータップの検出方法を提案している。まず、実験的に、キータップをした時の終点の3次元分布の中でキータップ、相関移動中の指先運動を含め、ユーザーの10本の指で空中タイピングパターンを調べた。結果に基づいて、実際にキータップをしている指（以下:Active Finger）とは別の指（以下:Passive Finger）も移動することが分かった。多くの人のPassive FingerとActive Fingerとの動いた距離のデータの、収集を行った。(図2.1)の割合でActive Fingerが移動すれば移動すればPassive Fingerも移動することが分かった。表の中は割合はAR(式2.1)である。Passive Fingerの移動距離も考えることにより高い確率で検出できるようになった。

$$AR = \frac{\text{相関する指の移動距離}}{\text{キータップの指の移動距離}} \times 100\% \quad (2.1)$$

| | | | | | |
|----------------|-----|-------|-------|-------|-------|
| Passive Finger | 小指 | 27.5% | 32.9% | 48.1% | |
| | 薬指 | 33.4% | 51.8% | | 60.6% |
| | 中指 | 46.3% | | 51.0% | 37.1% |
| | 人差指 | | 42.5% | 33.4% | 31.2% |
| | | 人差指 | 中指 | 薬指 | 小指 |
| Active Finger | | | | | |

図 2.1: ActiveFinger と Passive Finger の相関関係 (表内の数字は AR)

最後に、通常のキーボードで 60WPM 以上の 8 人の参加者に試作品の評価実験を行った。その結果、参加者の平均で 23.0 WPM を達成することができることを示した。また、後に精度を犠牲にすることなく、3 回の練習後に 29.2WPM を達成した。

2.3 携帯端末向け多指 AR タイピングインタフェース

樋口ら [4] の研究では、携帯端末の背面にカメラを設置しユーザーの手指を含む実画像を取得しオプティカルフローにより手指の動きを認識することで、携帯端末の画面上

に重畳した仮想キーボードで五本指による空中タイピングを可能にするインタフェースを可能にした。このインタフェースでは、オプティカルフローで手の認識を行っており、拡張現実感 (Augmented Reality: AR) の技術を用いることで、仮想物体の直感的な操作を実現している。動作テストから得られたタイピング時間の平均を1分あたりの入力単語数 (Words Per Minute: WPM)[10] に換算したところ、9.23[WPM] であった。

第3章 設計準備

本章は、提案手法を設計するために使用した装置 LeapMotion、キー配置、入力方式、開発環境の紹介をしていく。

3.1 LeapMotion

今回、手や指の解析を行う上で使用したセンサーは LeapMotion(図 3.1) である。LeapMotion は、2012 年に販売された手、指、およびペンなどを認識することができるデバイスである。認識精度は、1/100mm である。これはほかのモーションキャプチャデバイスよりも正確である。LeapMotion の大きさは、縦 30mm、横 80mm、高さ 11mm、重量は 45g で、とても小さいため、持ち運ぶ際にも便利である。



図 3.1: LeapMotion

Leap Motion では光学センサと赤外線カメラを使用しており、次のものを取得できる。

- 手
- 指
- ツール (棒状のもの)

それぞれに、位置や方向、速さや向きといったパラメーターを取得できる。認識範囲は、LeapMotion の中心を頂点とした逆ピラミッド形で、距離は約 25~600mm である。

3.1.1 手, 指を検出

LeapMotion は、手、指の位置、向きがわかる。手の一部分が表示されていない場合であっても、予測追跡をするために人間の手のモデルを使用している。手全体とそのすべ

ての指がはっきりと見えるときが最適であるが、手または指の一部が認識できていない場合であっても、常に5本指の座標を読み取ることができる。

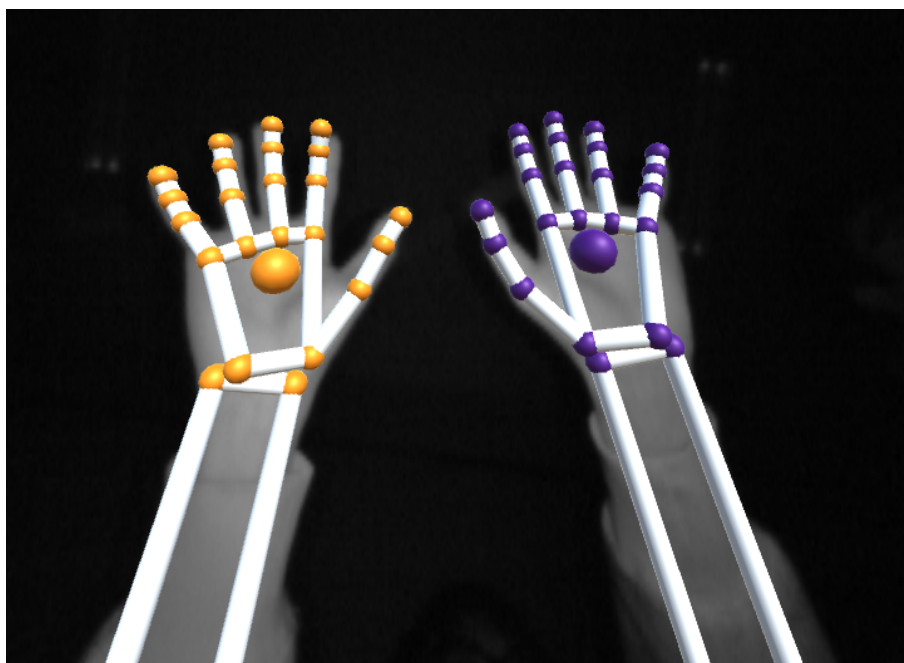


図 3.2: ビジュアライザーを用いて手を検出

3.1.2 ジェスチャーの検出

LeapMotion は備え付けでジェスチャーを認識できるシステムがある。認識できるジェスチャーは次の4つである(図 3.3)。

- サークル (TYPECIRCLE) : 円を描く動作
- キータップ (TYPEKEYTAP) : キーを押しているような(「下方向」にタップする)動作
- スクリーンタップ (TYPECREENTAP) : スクリーンを押しているような(「前方向」にタップする)動作

- スワイプ (TYPESWIPE)：指を伸ばした状態の手で直線を描く動作

このうち、今回の研究で使用したものは、キータップジェスチャーである。ジェスチャーを行った指は検出することができないが、ジェスチャーが起きた座標を取ることができる。

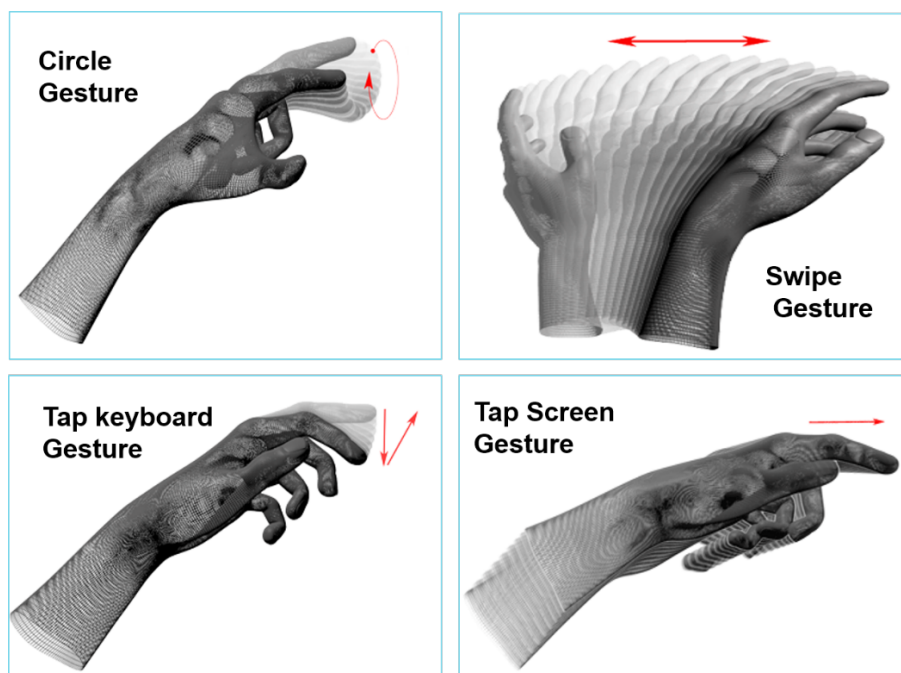


図 3.3: LeapMotion で検出できるジェスチャー (Leap Motion マニュアル [5] より)

3.1.3 座標

画面上での座標系 (図 3.4) と LeapMotion での座標系 (図 3.5) は、座標の向きは同じだが、原点の位置が異なる。また、LeapMotion の座標系と画面上での座標系では、Y 軸の向きが異なる。表示する際にその 2 つの点に気を付けて表示する必要がある。

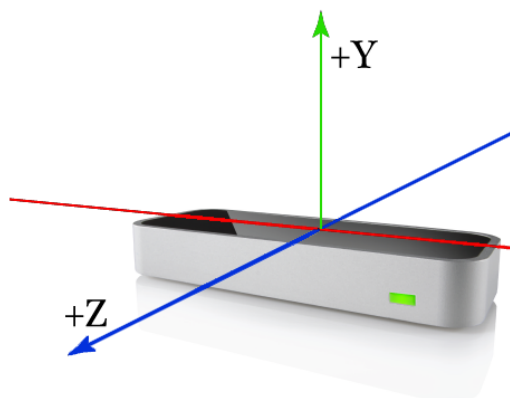


図 3.4: LeapMotion での座標軸



図 3.5: ディスプレイ上での座標軸

3.2 キーの配置と入力方法

一般的な物理キーボードの配置(図 3.6)として用いられている QWERTY 配列である。QWERTY 配列である理由は、英語をタイピングするときに最も使われるキーなどを放して配置することで 10 本の指をすべて活用でき、タイピング速度が向上できるといわれている。入力方法として、キートップを押し込むことにより文字が出力される。



図 3.6: 一般的なキーボードの配列 (QWERTY 配列)

3.3 開発環境

本研究において開発に用いた PC の詳細は (表 3.1) である。

表 3.1: 研究に用いた PC の詳細

| ・ノート PC | |
|-------------|-------------------------------|
| OS | : Windows7 Home Premium |
| プロセッサ | : Intel core i5-3317U 1.70GHz |
| システムの種類 | : 64 bit |
| 実装メモリ (RAM) | : 8.00GB |

今回開発には Eclipse を用いた。開発言語は Java である。。LeapMotion とのインターフェイスは、LeapMotion 社提供の LeapMotionSDK を利用した [6]。

第4章 提案手法の設計

この章では、提案手法の2つの設計を行っていく。設計する提案手法として、特殊な入力方式については考えず、既存の入力方式で設計することを考えている。この章では、システムの内容についての説明を述べる。

1つ目は、一般的なキーボードと同じように10本の指でキータップして入力する方法である。

2つ目は、手全体を下に下げることによって反応する一本指タイピングを考案した。ここでなぜ、二つ提案して比較を行うかについて説明する。一般的なキー入力においては、10本の指でタップし入力するタイピング方法が、普及している。しかし空中タイピングにおいて触覚的フィードバックがない状態では、ホームポジションやブラインドタッチ等のタイピング技術が、空中タイピングにおいて適用されない可能性がある。そのため、提案手法として2つ提案して比較しようと考えた。

4.1 提案手法1 (10本指キータップ入力手法)

今回、私が提案する手法は、視覚的フィードバックを含むタップ入力式キーボードを考案した。(以下提案手法1を(10本指キータップ入力手法とする。))物理キーボードにおいて、一般的なタイピング方式である。

画面上に、常に手の位置とキーボードを表示する。10本の指のどれかをターゲットのキートップに照準を合わせ、指タップのジェスチャーをすることで、そのキートップを打鍵したことにする。

指を認識してから文字を出力するまでを以下の手順で行った。

1. 10本の指の位置を常に取り得
2. キーボード、指の位置を表示
3. どの指でキータップが発生したかを判定
4. どのキーを打鍵したかを判別する
5. 文字を出力

今回の研究では、タイピング技術に優れていない人にも使いやすく考えているため、タイピングのホームポジションにとらわれずに、どの指でどのキーを押してもよいに考えている。

この節では、上記のシステムの構成にしたがって設計紹介を行っていく。

4.1.1 10本の指の位置を常にとる

LeapMotion は 1 秒間に 60 フレームを取ることができる。LeapMotion を用いて指の高さを計測した結果、1 タップにかかる時間 (図 4.1) は 0.2~0.3 秒 (12~18 フレーム) である。1 タップにかかる時間の座標は常に取り出せるようにしたいので、構成要素 24 個の配列を作成し新しいフレームの指の座標を入れていく。一番古いフレームを消していくことにより、0.4 秒前までの指の座標を常に取り出せるようにしている。

4.1.2 キーボード、指の位置の表示

今回指を表示するにあたって JAVA の GUI フレームワークである Swing を利用して指の位置を表示した。

表示するキーボードのタイプとして QWERTY 配列キーボード (図 4.2) を考えた。

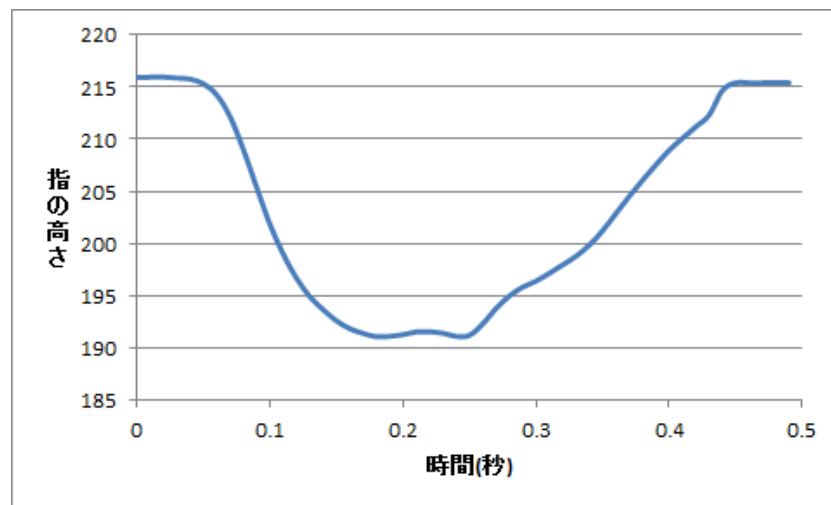


図 4.1: 人差し指をタップさせた時の時間と高さの関係

指の表示について、3.1.3 で述べたように座標の変換が必要である。LeapMotion で得られる Z 座標を GUI 上では Y 軸に変換する。また、GUI 上では負の数値はなく LeapMotion の座標では負の数値まで得られるため、各数値に 180 加えることにした。

指は、常に目立ちやすい赤の枠で丸表示することにした。また、タップした時には指は赤く塗りつぶし、押されたキートップは黄色にすることにより、実際にどのキートップを押したのかを使用者にわかりやすくした (図 4.3)。



図 4.2: 10 本指キータッパ入力手法に表示した QWERTY 配列キーボード



図 4.3: 人差し指で F を押したとき

4.1.3 どの指でキータップが発生したかを判定

(3.1.2) で述べたキージェスチャーでは、どの指がキータップをしたかを判別できない。そのため指の判別には次の条件を利用した。

- 指の移動量
- 高さが一番低い指がほかの4本の指の平均と比較
- キータップジェスチャーの座標との比較
- 高さ以外の移動量

上の4つの条件は後に述べる。

LeapMotionのキージェスチャーを合図として、キージェスチャーを行ったとき、上の4つのうち2つの認識した指が一緒であるなら、その指がキータップしたとする。

これにより各指のそれぞれで50回ずつキータップした時の認識率は、(表4.1)となった。10本の指の平均は84.8%となった。誤認識は、違う指を認識、指を認識しないの2つである。指を認識しないことを除外すれば、約95%であった。これは5文字の英単語を打つと仮定したら、全部を一度もミスタップをせずに入力できる確率は0.95の5乗で78%となる。

表 4.1: キータップの認識率

| 認識率 (%) | 小指 | 薬指 | 中指 | 人差し指 | 親指 |
|---------|----|----|----|------|----|
| 左手 | 80 | 84 | 88 | 94 | 78 |
| 右手 | 78 | 86 | 90 | 92 | 78 |

指の移動量

指の移動量は(式 4.1) で計算を行った。

$$dist = \sqrt{(ax - x)^2 + (ay - y)^2 + (az - z)^2} \quad (4.1)$$

ax,ay,az は 0.4 秒間の X 座標、Y 座標、Z 座標平均である。指の移動量が一番大きな値とその指の含まれる手で移動量が小さな指の値を取る。一番大きな値が一定値より小さいとき、手は静止しているとする。一番小さな値が一定値より大きいとき、他の指も移動しているということなので手全体が動いているとする。

一番低い指がほかの 4 本の指の平均と比較

キータップのジェスチャー時その手の中では一番低くなる。一番低い指の高さと他の指の平均との差の大きさでどの指がキータップしたかを検出する。

しかし、実装したところ、誤認識が多い結果になった。原因として、手の高さが違うためである(図 4.4)。もう片方の手より高い位置の手にある指でキータップした時、認識しない。それを修正するために、一度それぞれの手で一番低い指の高さと指の平均の高さの差を取り、右手と左手で比べ差が大きい方がキータップした指とした。



図 4.4: 手の位置が一緒ではない(右手の方が高い)

キータップジェスチャーの座標との比較

(3.1.2)で述べたように LeapMotion のジェスチャー機能の座標を用いた。キータップした場所の x 座標値と 10 本指の指先の x 座標値との差を比較して、差が小さい指をキータップした指だと判定する。 z 座標値を使用しなかった理由は、 z 軸は奥行きを指してい

るため、左右の手の各指先の z 座標値は各指ごとに近い値を示す。キータップが反応したとき 10 本指の z 座標値を比較すると近い値が 2 つ以上存在するからである。y 軸を使用しなかった理由としては手の高さがそれぞれで違うとき (図 4.4)、高さとの差を取ると誤認識した指を取ることがあったからである。

高さ以外の移動量

小指と親指のタップ動作を検出する際に、高さだけでは認識されないという問題が出た。その理由として、物理的フィードバックのない状態で小指と親指はタップさせたときに高さの値があまり変化しないからである。高さを意識してタイピングするとどうしても指が不自然な動作になる。しかし、高さ (Y 座標) 以外の X,Z 座標は、自然な形でタイピングを行うと、変化することが実験で得られた (図 4.5) (図 4.6)。この結果より X,Z 座標の移動量 (式 4.2) の最大を取ることにより小指と親指の認識精度を上げることに成功した。

$$dist = \sqrt{(ax - x)^2 + (az - z)^2} \quad (4.2)$$

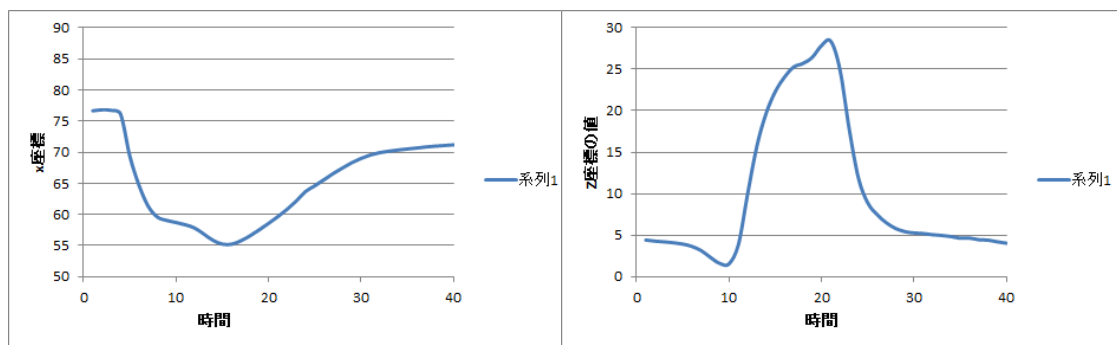


図 4.5: 小指をタップ動作させたときの高さ (Y 座標) 以外の X 座標 Z 座標の変化の量

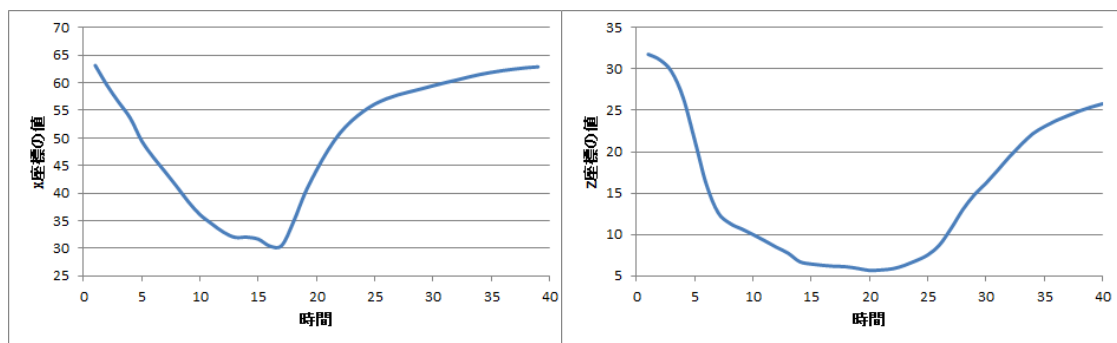


図 4.6: 親指をタップ動作させたときの高さ (Y 座標) 以外の X 座標 Z 座標の変化の量

4.1.4 どのキートップを打鍵したかを判別

どのキートップを打鍵したかの判別には、キートップの中心との距離で判別している。キータップが発生したとする指の座標の x 座標 z 座標を取り、キートップの中心の座標とを三平方の定理にて距離を計算し、その距離が一番小さいものが打鍵されたキートップであるとした。

しかし、空中でキータップをする際に、触覚的フィードバックがないことにより、強くキータップをしてしまう場合もある。強くキータップを行うと、x 座標 z 座標まで変化してしまい隣のキートップが反応してしまうという問題点 (図 4.8) が発生した。

その問題点を解決するため 0.4 秒前までのフレームでキータップが発生した指の高さの値が最も大きいフレームを目標とするキートップに照準があっているときということを考えた。そのフレームの指の x 座標と z 座標との距離を取った。これにより強く打鍵してしまっても打鍵する人の意図したキートップを押すことができるようになった。

4.1.5 文字を出力

押されたときキーの情報を出力するようにした。情報の内容として、色の変化、キーコードとなっている。(4.1.2) で述べたようにキーが押されたとき黄色になり、0.8 秒後に白に戻すという動作を入れた。また、実際に物理的キーが押された現象を再現するには、

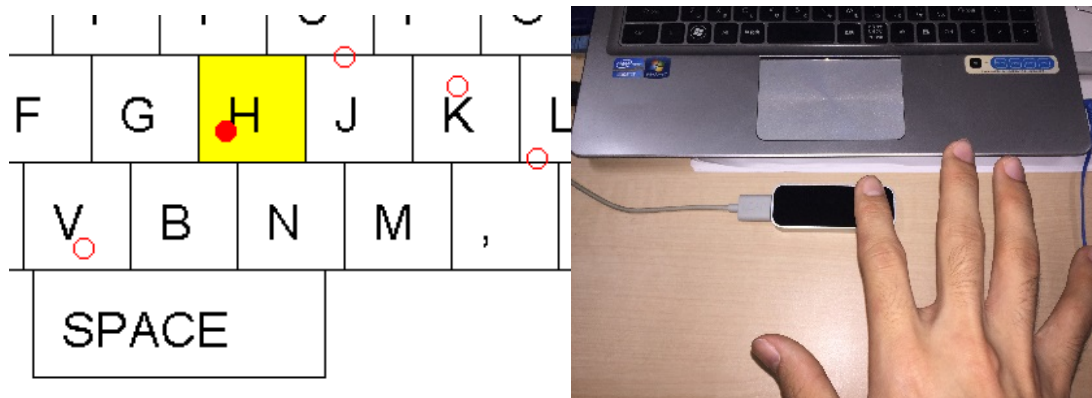


図 4.7: 弱めにタップの動作を行った場合

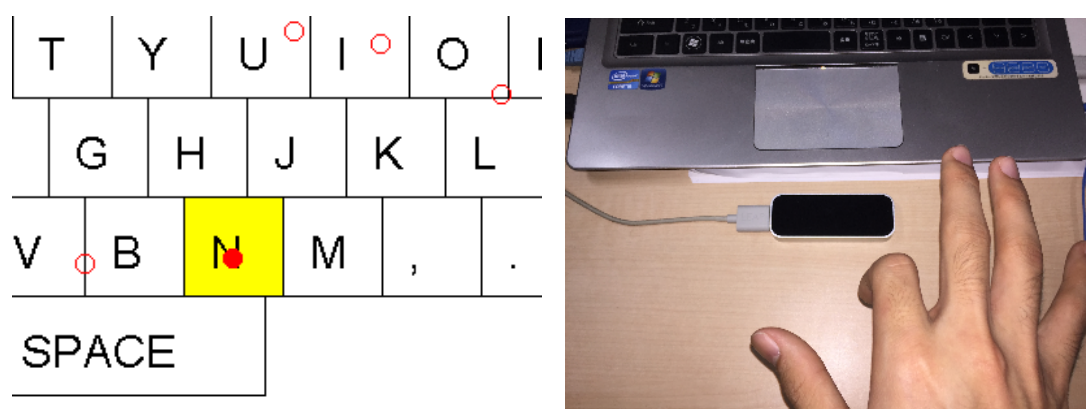


図 4.8: H に照準をあわせたが強めにタップの動作を行ったため N が打鍵された

java.awt.Robot クラスを用いた。java.awt.Robot クラスは、テストの自動化、自動実行のデモ、およびマウスやキーボード制御が必要なアプリケーションのために、ネイティブなシステム入力イベントを生成することができる。java.awt.Robot クラスに打鍵されたとするキーコードを渡し、文字を出力する。

打鍵された後に物理的キータップの音を再生するようにし、聴覚的フィードバックを含ませた。

実際に実装してみて、キータップが行われる動作中に連続して文字が打鍵された。それを解決するために、一般的なタイピングスピードであるパソコンスピード検定 [7] の 2 級 (表 4.2) を目安に 1 回タイピングされたら 0.4 秒間文字を打鍵されないように調整した。

表 4.2: パソコンスピード検定 [7]

| | 5 級 | 4 級 | 3 級 | 2 級 | 1 級 |
|-------------|--------|--------|--------|--------|---------|
| 文字数 (10 分間) | 100 文字 | 300 文字 | 500 文字 | 700 文字 | 1000 文字 |

4.2 提案手法 2 (一本指タイピング手法)

2 つ目の提案手法として、我々は手の動きで打鍵する方法 (以下、一本指でタイピング手法) を考えた。物理キーボードに慣れていない人が最初のころに行うであろうタイピング動作 (図 4.9) である。

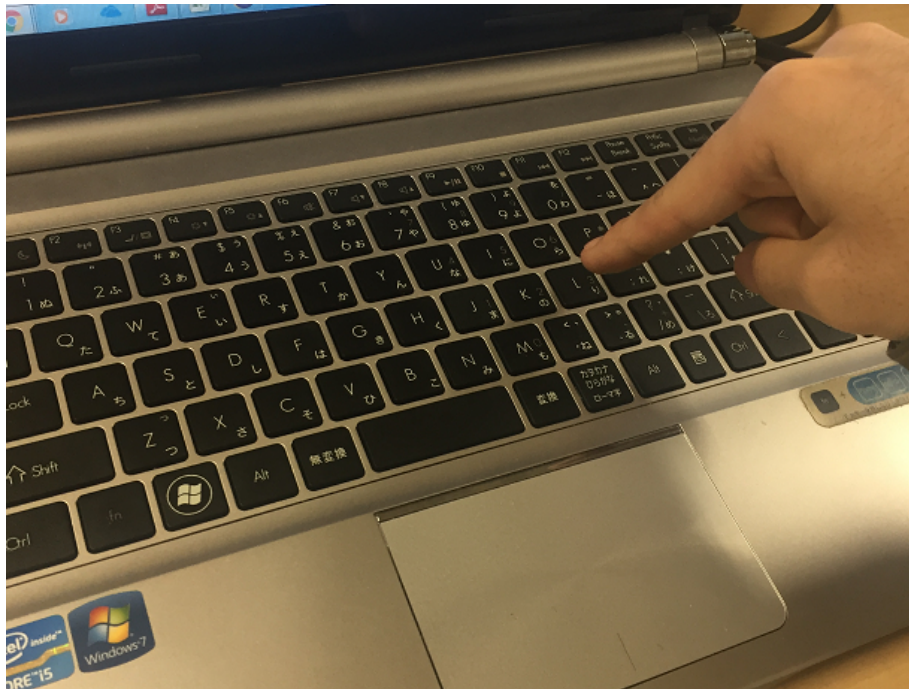


図 4.9: 物理的キーボードで一本指のタイピングする様子

一本指タイピング手法では、一つの指を下に下げることによって指が赤くなり、その指が含まれる手の全体を下げることにより打鍵したとする。その手の中で一番座標の低い指の値を取り、どのキートップが打鍵されたかを判定するものである。

指を認識してから文字を出力するまでを以下の手順で行った。

1. 10本の指の位置を常に取得
2. キーボード、指の位置を表示
3. ターゲットの指を判別
4. 手の動きにより打鍵したかを判別
5. どのキーを打鍵したかを判別

6. 文字を出力

1,2,5,6については10本指キータツプ手法と一緒にある。この節では、10本指キータツプ手法とは違う手順3,4の設計説明を行っていく。

4.2.1 ターゲットの指を判別

ターゲットの指の検出には、LeapMotionで認識することができる手のひらの座標を率いた。手のひらの中心と指の高さ差が大きい指を取る。高さの差一定値より大きいときその指をターゲットの指とした。

4.2.2 手の動きにより打鍵したかを判別

手の下がりによって認識することになると、疲労を感じて手が下がったりすると、誤認識することが分かった。そこで前0.1秒間の手の下がりの勢いによって検出することにした(図4.10)。手の勢いの計算方法として、高さの変化の積分を利用した(式4.4)。

$$\text{手の勢い} = \int_0^{0.1} (t \text{ 秒の時の高さ}) dt \quad (4.3)$$

$$\doteq \sum_{t=0}^{0.1} (\text{平均の高さ}) - (t \text{ 秒前の高さ}) \quad (4.4)$$

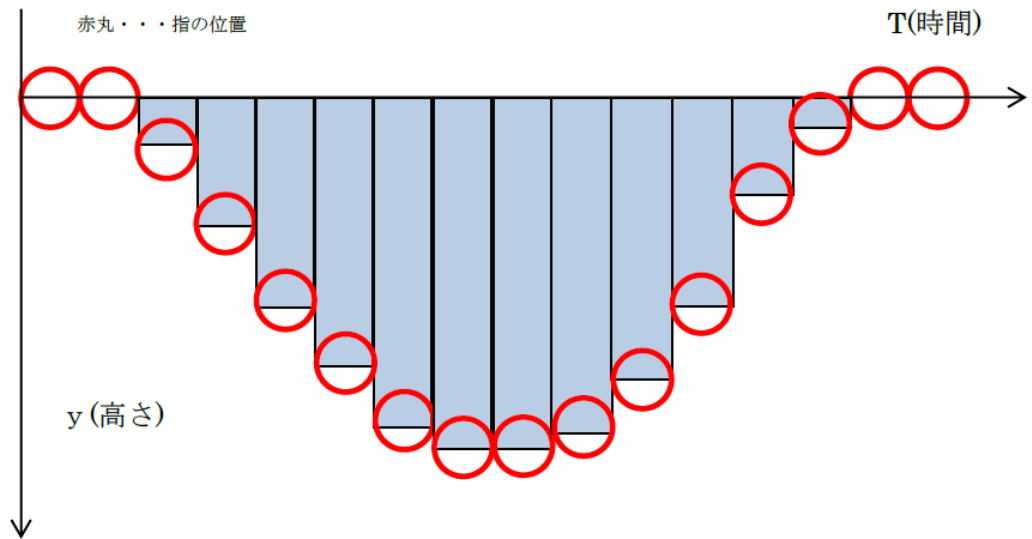


図 4.10: 手を下に押し込んだ時の図

第5章 評価実験

本章では、前章までで設計した提案手法について実際に実装をして、評価と比較を行っていく。

5.1 実験目的

10本指キータップ手法と1本指タイピング手法で入力速度と精度を計測するために被験者実験を行った。入力速度と精度(正誤率)の計算方法は(式5.1)と(式5.2)で計算を行った。(式5.1)の計算式に1/5を掛けている理由として入力速度WPM(略:words per minutes)は5文字で1単語(words)と仮定しているからである。実験の結果よりそれぞれの入力速度と精度を計測と比較をして、評価していく。

また、実際の物理的キーボードにおいて、10本指でタップ式タイピング(10本指キータップ手法)することは練習を行えば行うほど早くなる。しかし、1本指タイピング(1本指タイピング手法)はあまり練習しても効果が得られないである。そこで、今回10本指キータップ手法と1本指タイピング手法を同じ条件でそれぞれ行った後、練習してもう一度実験を行ってみようと考えた。これは提案手法が、少し慣れれば精度、入力速度が向上するかを確かめるためである。

$$WPM = \frac{(\text{打鍵したすべての文字数}) - (\text{ミスした文字数})}{\text{かかった時間(秒)}} \times 60(\text{秒}) \times \frac{1}{5} \quad (5.1)$$

$$\text{正誤率} = \frac{(\text{打鍵したすべての文字数}) - (\text{ミスした文字数})}{\text{打鍵したすべての文字数}} \quad (5.2)$$

5.2 実験方法

被検者として、20代男性の4名に対して行った。順序効果をなくすため、10本指キー
タップ手法から行う2人と1本指タイピング手法から行う2人に分けた。

被験者には、2つの提案手法で以下の6つの文、単語を入力してもらった。

1. 生年月日 (8文字)
2. Leapmotion (10文字)
3. soccer (6文字)
4. baseball (8文字)
5. the quick brown fox jumps over the lazy dog. (44文字)
6. pack my box with five dozen liquor jugs (41文字)

1は自分の生年月日を数字で入力してもらおうというものである。5と6は全アルファ
ベットが入っている文になっている。それぞれの提案手法の入力の前に、説明を含む10
分間練習を行ってもらった。それぞれ終わった後に、練習してもらい40分後にもう一度
入力してもらった。時間の計測については、プログラムを作成し1回目のEnterでタイ
マースタート2回目のEnterでタイマーストップとした。また、精度を計測するために1
語入力されるたびにカウントをし、BSを押した回数でミスをしたとしミスカウントを増
やしていくプログラムに含ませた。また、最後にアンケートとして、実際に使用した感
想を聞いた。

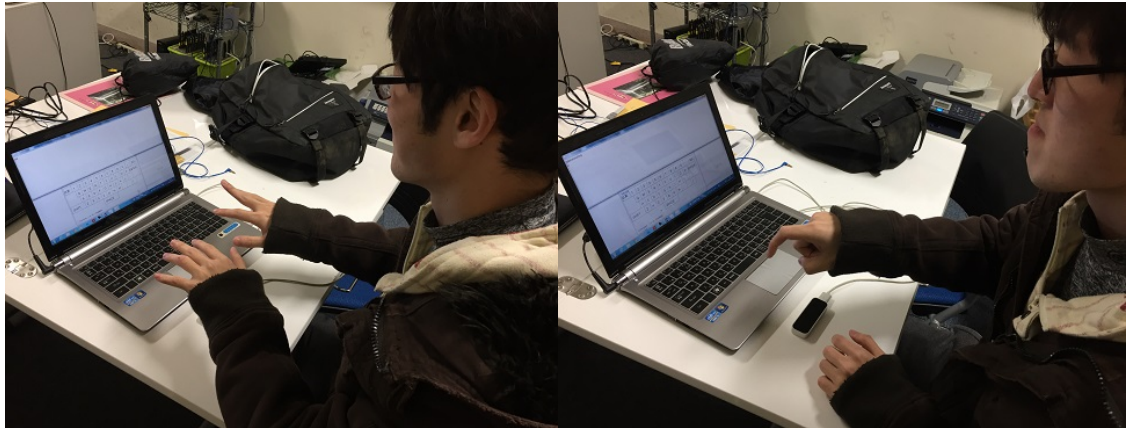


図 5.1: 10 本指キータツプ手法を行っている様子
図 5.2: 一本指タイピング手法を行っている様子

5.3 実験結果

実験結果は、(表 5.1)～(表 5.6) に示す。被験者それぞれの実際の物理キーボードでの入力技術として、AB はブラインドタツプできない人約 20WPM、CD はブラインドタツプできる約 40WPM である。(表 5.6) より、1 回目では入力速度は、10 本指キータツプ手法より一本指タイピング手法の方が速かった。2 回目では、10 本指キータツプ手法より一本指タイピング手法の方が速かったが、1 回目より差は小さくなっていった。1 回目と 2 回目では、10 本指キータツプ手法はとても早くなっていた。しかし、一本指タイピング手法では、あまり変わらないという結果になった。(表 5.5) より、一本指タイピング手法の方が 10 本指キータツプ手法より正確であった。正誤率は、単語では正誤率は上がるが、長文ではとても下がることになった(表 5.7)。それを踏まえて単語だけの WPM を計算しなおしてみた(表 5.8)。

表 5.1: 10 本指キータップ手法 (1 回目)

| 入力時間 (秒) | 被験者 A | 被験者 B | 被験者 C | 被験者 D |
|--|--------|--------|--------|--------|
| 生年月日 | 45.31 | 26.59 | 21.52 | 18.52 |
| LeapMotion | 35.79 | 46.0 | 15.89 | 20.65 |
| Soccer | 20.21 | 54.65 | 16.51 | 18.89 |
| baseball | 20.31 | 32.07 | 22.73 | 19.71 |
| the quick brown fox jumps over the lazy dog. | 285.93 | 164.07 | 132.04 | 112.04 |
| pack my box with five dozen liquor jugs | 175.51 | 156.26 | 154.41 | 95.41 |

表 5.2: 10 本指キータップ手法 (2 回目)

| 入力時間 (秒) | 被験者 A | 被験者 B | 被験者 C | 被験者 D |
|--|--------|--------|-------|--------|
| 生年月日 | 12.51 | 31.01 | 11.93 | 11.51 |
| LeapMotion | 13.47 | 42.11 | 13.77 | 14.49 |
| Soccer | 7.06 | 50.02 | 9.51 | 12.51 |
| baseball | 12.00 | 30.1 | 15.31 | 15.52 |
| the quick brown fox jumps over the lazy dog. | 103.03 | 152.35 | 95.73 | 115.01 |
| pack my box with five dozen liquor jugs | 95.00 | 125.01 | 89.09 | 96.14 |

表 5.3: 一本指タイピング手法 (1 回目)

| 入力時間 (秒) | 被験者 A | 被験者 B | 被験者 C | 被験者 D |
|--|-------|-------|-------|-------|
| 生年月日 | 31.82 | 15.31 | 14.03 | 17.13 |
| LeapMotion | 11.22 | 15.91 | 12.31 | 19.19 |
| Soccer | 6.44 | 14.76 | 14.63 | 15.98 |
| baseball | 13.31 | 17.02 | 18.81 | 19.85 |
| the quick brown fox jumps over the lazy dog. | 88.08 | 95.76 | 76.61 | 86.61 |
| pack my box with five dozen liquor jugs | 57.09 | 85.84 | 61.14 | 76.12 |

表 5.4: 一本指タイピング手法 (2 回目)

| 入力時間 (秒) | 被験者 A | 被験者 B | 被験者 C | 被験者 D |
|--|-------|-------|-------|-------|
| 生年月日 | 23.7 | 24.72 | 12.20 | 16.34 |
| LeapMotion | 31.81 | 15.91 | 10.70 | 20.87 |
| Soccer | 13.25 | 17.09 | 14.31 | 15.26 |
| baseball | 15.52 | 18.01 | 23.31 | 19.9 |
| the quick brown fox jumps over the lazy dog. | 79.53 | 88.08 | 81.11 | 89.33 |
| pack my box with five dozen liquor jugs | 57.35 | 57.09 | 57.95 | 82.95 |

表 5.5: 正誤率

| 正誤率 (%) | 被験者 A | 被験者 B | 被験者 C | 被験者 D |
|---------------------|-------|-------|-------|-------|
| 10 本指キータップ手法 (1 回目) | 76.00 | 76.88 | 81.60 | 79.17 |
| 10 本指キータップ手法 (2 回目) | 90.48 | 76.88 | 87.5 | 86.36 |
| 一本指タイピング手法 (1 回目) | 93.66 | 89.86 | 91.72 | 91.72 |
| 一本指タイピング手法 (2 回目) | 89.86 | 84.17 | 96.38 | 95.68 |

表 5.6: 文字入力速度 (WPM)

| 入力速度 (WPM) | 被験者 A | 被験者 B | 被験者 C | 被験者 D |
|---------------------|-------|-------|-------|-------|
| 10 本指キータック手法 (1 回目) | 2.73 | 3.32 | 4.91 | 4.28 |
| 10 本指キータック手法 (2 回目) | 6.57 | 3.46 | 6.78 | 6.02 |
| 一本指タイピング手法 (1 回目) | 7.76 | 6.52 | 8.79 | 6.544 |
| 一本指タイピング手法 (2 回目) | 7.67 | 6.74 | 8.00 | 6.52 |

表 5.7: 正誤率 (入力してもらった分野ごと)

| 正誤率 (%) | 被験者 A | 被験者 B | 被験者 C | 被験者 D |
|---------------------|-------|-------|-------|-------|
| 1 ~ 4 (単語) | | | | |
| 10 本指キータック手法 (1 回目) | 80.70 | 71.88 | 85.19 | 85.19 |
| 10 本指キータック手法 (2 回目) | 97.87 | 75.41 | 95.83 | 92.00 |
| 一本指タイピング手法 (1 回目) | 90.20 | 95.83 | 93.88 | 93.88 |
| 一本指タイピング手法 (2 回目) | 93.88 | 80.70 | 100 | 97.87 |
| 5, 6 (長文) | | | | |
| 10 本指キータック手法 (1 回目) | 73.73 | 79.82 | 79.81 | 76.32 |
| 10 本指キータック手法 (2 回目) | 87.00 | 78.38 | 83.65 | 83.65 |
| 一本指タイピング手法 (1 回目) | 95.60 | 87.00 | 90.63 | 90.63 |
| 一本指タイピング手法 (2 回目) | 87.89 | 86.14 | 94.57 | 94.57 |

表 5.8: 文字入力速度 (WPM) (短い単語のみ)

| 入力速度 (WPM) | 被験者 A | 被験者 B | 被験者 C | 被験者 D |
|---------------------|-------|-------|-------|-------|
| 10 本指キータップ手法 (1 回目) | 4.54 | 3.46 | 7.02 | 7.10 |
| 10 本指キータップ手法 (2 回目) | 12.56 | 3.60 | 10.93 | 10.22 |
| 一本指タイピング手法 (1 回目) | 8.03 | 8.76 | 9.23 | 7.65 |
| 一本指タイピング手法 (2 回目) | 8.79 | 6.24 | 9.12 | 7.62 |

5.4 考察

一本指タイピング手法の方が知識がなくても使いやすいとわかった。理由として入力の方法の容易さにあると考えられる。一本指タイピング手法は一本の指で狙って押し込むだけである。使用者にとって、10 本指キータップ手法での空中でのキータップの動作と常に 10 本指の位置を画面からえるのが難しいことが考えられる。触覚的フィードバックがないことによって押したかわかりにくく、強くキータップしてしまう。

物理キーボードの入力技術が、10 本指キータップ手法にも有用ではないと結果が得られた。物理キーボードとの速度に差が生じた理由としてブラインドタッチができないことが理由である。そのことにより、物理キーボードではホームポジションをうまく 10 本指でタイピングできていたが、10 本指キータップ手法ではうまく 10 本の指をつかえていない印象であった。

被験者 B に関してほかの被験者より精度と速度がともに低かった。Leap Motion 付属のアプリ「ビジュアライザー」を見ながらキータップしてもらった結果、中指を動かしても薬指しか動いていないという事象が起きていた。

感想で一番多く寄せられたものは疲労感であった。長文と単語では入力語数と長時間による手の疲労などにより、差が大きく生まれたと考えられる。特に 10 本指キータップ

手法では、その差は大きくなった、要因の一つとして10本の指の向きがばらばらであったことが挙げられる。実験を見ていて、今回仮想キーボードの位置として机の上にあるように仮想した。しかし、長文になると徐々にパソコンの画面上にキーボードがあるように手が移動しているのが見られた(図5.3)。

今回キーボードの大きさを固定していたことも精度や速度が下がっていた理由であった。人により手の大きさが違うので、その人の手の大きさを踏まえてキーボードを大きくすれば精度速度は向上すると考えられる。

10本指キータップ手法では2回行った結果二回目の方が精度が大きく上がっている。これより入力速度と精度は向上すると考えられる。精度では10本指キータップ手法(2回目)は一本指タイピング手法より、大きく劣っているが、速度に関してはほぼ差がないことを考えると、この先もっと練習を踏まえば速度に関して10本指キータップ手法の方が一本指タイピング手法より早くなることが考えられる。あくまで、参考でしかないが筆者の精度と入力速度は(表5.9)となった。実際に練習をすればもっと向上できるのではないかと考えることができる。



図 5.3: 画面をキータップしているかのような動き

表 5.9: 筆者のデータ

| | 10本指キータップ手法 | 一本指タイピング手法 |
|------------|-------------|------------|
| 入力速度 (WPM) | 12.32 | 10.11 |
| 正誤率 (%) | 95.83 | 94.57 |

第6章 結論

今回 10 本指キータップ手法と一本指タイピング手法の設計と比較を行った。一本指タイピング手法の方が練習なしでは、すぐに使えるという利点が得られた。その理由として、入力方式の容易さ、指の位置を目で追う数の違いがある。10 本指キータップ手法では練習をすれば、入力速度が向上することが分かった。10 本指キータップ手法 (2 回目) は一本指タイピング手法より、練習を踏まえれば速度に関して 10 本指キータップ手法の方が一本指タイピング手法より速くなることが考えられる。

視覚的フィードバックを含む空中タイピングの開発としては、一本指タイピング手法の方が使いやすく、誰にでも使いやすいを限られない。

視覚的フィードバックを含む影響として、10 本の指でタイピングする 10 本指キータップ手法は 10 本の指を目で追う必要があるため、物理キーボードのタイピング技術が空中タイピングの技術として使うことができないというのが、一番の影響である。

短い単語レベルでは物理キーボードとの差はあるが、予測変換を用いればあまり感じなくなるであろう。VR での入力などにはよい手法だと考えられる。

また、10 本指キータップ手法は、キータップの配置や指の位置を常に見て、文字を入力しないと行けない。Airkey[2] や ATK[3] の前の段階でキータップの配置を覚えていない人やブラインドタッチができない人のためには、10 本指キータップ手法は、有効な手段であると思う。

今後の課題として、精度の向上を上げられる。今回正誤率はミス入力の回数で計算をしているので数値には入っていないが、実際にはキータップをしても反応がしないこと

があった。これは使用者にとってストレスにつながるとともに、入力速度の低下にもつながる。今後、キータップの精度をより向上できる研究を期待している。

謝辞

本研究を完成するにあたり、ご指導くださりました三浦准教授にお礼申し上げます。加えて、サーベイ輪講・中間発表で御教授していただいた情報セクションの先生方、先輩方に御礼申し上げます。また、研究室の方々を始めとする、実験に協力していただいた方々に、重ねて御礼申し上げますから感謝申し上げます。

参考文献

- [1] SonyInteractiveEntertainmentInc. PlayStationVR. http://www.jp.playstation.com/psvr/index.html?hfclick=HDR_psvrtopp, 2016.
- [2] 星野元気. Leapmotioncontroller による 空中仮想キーボード AirKey の開発. 2014 年九州工業大学卒業論文, October 2014.
- [3] Mingrui Zhang Sida Gao Ke Sun Yuanchun Shi Xin Yi, Chun Yu. Latk: Enabling ten-finger freehand typing in air based on3d hand tracking data. In *ACM User Interface Software and Trchnology Symposium*, October 2014.
- [4] 樋口政和, 小室孝. 高速手指動作認識による携帯端末向け多指 AR タイピングインタフェース. 2015.
- [5] Leap Motion Inc. Gesture — Leap Motion Java SDK v3.2 documentation. <https://developer.leapmotion.com/documentation/java/api/Leap.Gesture.html>.
- [6] Leap Motion Inc. LeapSDK. <https://www.leapmotion.com/setup>.
- [7] 日本情報処理検定協会. パソコンスピード認定試験. http://www.goukaku.ne.jp/test_pcspeed.html, 2016.