

2016 年度 卒 業 論 文

ジェスチャー操作とグラフィティ操作による
簡易的な文字入力方法の提案

指導教員：渡辺 大地 講師

メディア学部 ゲームサイエンスプロジェクト
学籍番号 M0113081
大西 未来

2017 年 3 月

2016 年度 卒 業 論 文 概 要

論文題目

ジェスチャー操作とグラフィティ操作による
簡易的な文字入力方法の提案

メディア学部

学籍番号：M0113081

氏
名

大西 未来

指導
教員

渡辺 大地 講師

キーワード

文字入力、ジェスチャー操作、グラフィティ文字、
Leap motion、ハウズドルフ距離

現在使われているソフトウェアキーボードは、タイピングに支障が出てしまう恐れがあったり、文字入力時に周りの影響を受けやすいという問題点がある。本研究の目的は、文字入力の弊害を減らす両手を使ったジェスチャー操作方法を実現する。大人子供でも使える文字入力方法を提案し、入力がスムーズに行えるようにする。入力デバイスとして Leap motion を使用する。先行手法の問題点である操作の難しさやバッティングといった誤認識の多さを解決することである。文字入力手法は、ジェスチャー認識とグラフィティ入力を組み合わせる方式を採用した。ジェスチャー認識は、Leap motion SDK と呼ばれる開発ツールを使用した。SDK に予め提供されているサークルやスワイプ等の比較的に分かりやすい動作を認識し、文字入力の補助を行った。グラフィティ入力は、「グラフィティ文字」と呼ばれるアルファベットを簡略した一筆書き文字である。これは、指定した軌跡を描くことで、文字入力を行う手法である。操作の流れとして右手ではグラフィティによる文字入力を行い、左手で文字の消去等補助的な動作を行ってもらう。グラフィティ文字入力を行ってもらう際、本研究では動きをパターン化した。操作と数値データがどれくらい合致しているか、その一致度を測る。それを測るために図形の類似度を測定するために使われるハウズドルフ距離を用いた。本研究ではジェスチャー操作とグラフィティ文字入力操作の2つの入力方法を取り入れた。そしてそれぞれの認識時間の測定と、動作を20回程繰り返してもらいどれくらい正しく認識出来たかの認識率を検証した。また両手を使った入力方法の測定も行った。その結果として認識の精度は課題が残る結果となった。入力の動作に関してや誤認識に対して改善が必要であり、本研究の両手を使った入力方法も課題が残る結果となった。

目次

第1章	はじめに	1
1.1	研究背景と目的	1
1.2	本論文の構成	3
第2章	研究手法	4
2.1	手法説明	4
2.2	両手の役割	5
2.3	ジェスチャー操作	5
2.4	グラフィティ文字入力操作	6
2.5	グラフィティ文字入力の方法	7
2.6	両手を使った入力方法の説明	11
第3章	検証内容と結果	13
3.1	検証内容	13
3.2	ジェスチャー操作の検証方法と結果	13
3.2.1	ジェスチャー操作、検証の分析	15
3.3	グラフィティ文字入力の認識検証と結果	16
3.3.1	グラフィティ文字入力操作、検証の分析	18
3.4	両手を使用した時の認識検証と結果	19
3.4.1	両手を使用した認識、検証の分析	20
第4章	終わりに	21
	謝辞	23
	参考文献	24

目 次

2.1	SDK ジェスチャー例	5
2.2	LeapSDK 動作中	6
2.3	グラフィティ文字例	7
2.4	正しい手の形	7
2.5	認識可能状態	8
2.6	グラフィティ文字 A	9
2.7	グラフィティ文字 B	9
2.8	グラフィティ文字 C	9
2.9	グラフィティ文字 D	9
2.10	ハウズドルフ距離の例	10
2.11	ハウズドルフ距離の求め方 (グラフィティ文字 A の場合)	11
2.12	フローチャート	12
3.1	グラフィティ文字入力:検証中	16
3.2	左手:ジェスチャー操作	19
3.3	右手:グラフィティ文字入力操作	19
4.1	両手の認識	21

表 目 次

3.1	ジェスチャー認識時間 (秒)	14
3.2	ジェスチャー操作の成功率 (前半 10 回)	14
3.3	ジェスチャー操作の成功率 (後半 10 回)	15
3.4	ジェスチャー操作の成功率 (20 回)	15
3.5	グラフィティ文字認識時間 (秒)	16
3.6	グラフィティ文字入力の成功率 (前半 10 回)	17
3.7	グラフィティ文字入力の成功率 (後半 10 回)	17
3.8	グラフィティ文字入力の成功率 (20 回)	17
3.9	4つのグラフィティ文字が表示された時間	18
3.10	両手を使った入力の成功率	20

第 1 章

はじめに

1.1 研究背景と目的

現在パソコンなどの電子機器に使われている、文字入力を行うための一般的なデバイスはキーボードである。キーボードという物はパソコン等の電子機器に使われており、文字入力を中心とした機能を操作する入力デバイスの 1 つである。またタッチパネルデバイスで、疑似的に・画面上にキーボード機能を再現する「ソフトウェアキーボード」[1] という技術もある。

ソフトウェアキーボードは、物理キーボードを PC 等の画面上に再現したものである。物理キーボードを使わないで、文字入力を行える機能の事を指す。このソフトウェアキーボードが使われている場面として一般的な例では、スマートフォンやタブレット PC 等日常で使われている物が挙げられる。また、他の場面では、障害者の支援 [2] の一つとして使われている。ソフトウェアキーボードは物理キーボードの様に素早いタイピングによる入力ではなく、マウス操作やタッチ操作による、ポインティングが主な入力方法となっている。このキーボードには 2 つの大きな特徴がある。第 1 の特徴としてキー配置を変更し、自分好みにカスタマイズする [3] と言ったカスタマイズ性に優れている点である。第 2 の特徴は物理キーボードの様に素早い入力方法を行う方法として、スマートフォンの「フリック入力」といった片手での素早い文字入力が可能という点

である。ソフトウェアキーボードには上記のような利点がある。

ソフトウェアキーボードはアミューズメント施設のような場面で使用すると、照明等周りの影響を受けると使い辛い、という理由により使えない状況がある。本研究はそのようなソフトウェアキーボードが使えない場面での文字入力の手助けをする事を目的とする。ソフトウェアキーボードを用いずに入力を行う手法を提案する。その方法としてモーションセンサーデバイスを使用し、手を使って行うジェスチャー入力に着目した。その先行研究として、藤井ら [4] によるペン型のデバイスを用いた空中手書き認識がありそれを実現している。これは Web カメラをペンに取り付けて方向成分情報を読み取り、文字入力を行うシステムである。しかし、誤認識や処理時間の長さが問題点としてある。

また、細野ら [5] は Leap motion[6] と呼ばれるモーションセンサーデバイスを使用し、検知される両手の指の本数や動作の違いから文字入力方法を行う手法を提案した。細野らの手法はモーションセンサーデバイスを使用し、独自に開発したジェスチャー操作で文字を入力していくというものである。細野らの研究は「Leap motionSDK」 [7] を使用している。しかし、細野らの手法は操作感が悪いという事と、SDK によるジェスチャーと自分で作成したジェスチャー同士がバッティングし誤認識を引き起こすという問題を論文中に示している。

ジェスチャー認識をインターフェースとして利用する研究例は多数存在し、空中手書き文字認識を行う研究 [8] や、川上らの文字の認識を向上させようと手の認識を用いた研究 [9] が先行研究としてある。しかし認識率の低さ、誤認識についてが問題として論文中に示している。保呂らが複雑な方法を持ち出し認識率の向上を図り [10]、認識率は 95 %以上という高い認識率が実験から検出したと論文中に示している。保呂らの研究は認識の問題は無いとしても、文字を認識させる為に大きな準備が必要という問題点がある。星野らの論文ではカスタマイズ性が高い事 [11] を論文中では実証した。星野らの研究は空中に仮想のキーボードを作り入力を行う手法を提案した。この手法は問題点として入力に時間がかかる事と空中でキーボード入力を行うのは労力がかかる

問題点が評価として挙げられていた。先行研究の問題点を踏まえ、大人子供でも使える文字入力方法を提案する。

本研究では文字入力の問題を解決すべくジェスチャー操作とグラフィティ文字入力操作の2つの入力方法を取り入れた。グラフィティと呼ばれる文字入力の方法ならば素早くかつ簡単に文字入力が行えると仮定を立てた。そしてそれぞれの認識時間と、動作を20回程繰り返してもらいどれくらい正しく認識出来たかの認識率を検証した。また、両手を使った認識を行った。その結果、入力方法としては課題が残る結果を検出した。入力動作や認識に対して改善が必要であり、課題が残る結果となった。

1.2 本論文の構成

本論文の構成を説明する。第2章では本研究の手法を説明する。ジェスチャー操作とグラフィティ文字入力操作の2つと両手を使った入力方法についてを論じる。第3章ではそれぞれの認識時間と認識率を検証した。そして第4章にはその結果から分かった事を終わりとしてまとめる。

第 2 章

研究手法

この章は文字入力に関する手法の説明をする。ジェスチャー操作とグラフィティに関する説明と、そしてそれぞれの手法を論じる。またその入力方法に関連したアルゴリズムを示す。最後に両手を使った際の操作の流れと紹介を本章では説明する。

2.1 手法説明

今回の研究の目的である文字認識システムを実現する上で、本研究では 2 つの操作方法を使用した。1 つ目は関連研究でも紹介した細野らのやり方を参考にした「ジェスチャー操作」を使用した。そしてもう 1 つは一筆書き文字入力とも言われている「グラフィティ文字入力操作」を取り入れた。それぞれの役割や、グラフィティ文字入力については後述する。開発ツールとして Unity[12] を使用した。また入力機器として、モーションセンサデバイスである Leap motion デバイスを使用した。高星ら [13] も述べているように、Leap motion は赤外線 LED とカメラを用いて人の手や指の位置・姿勢を検出し、24 点のボーン情報を取得する事が出来る。繊細なデバイスではあるが、ジェスチャー動作を読み取るためのデバイスとして本研究に採用した。また近年 HMD(ヘッドマウントディスプレイ) による VR(ヴァーチャルリアリティ) エンタテインメント

が注目を浴びており、それに関する岩谷らの研究 [14] がある。しかし、本研究は VR の事を念頭には置いておらず HMD は使用しない。ジェスチャー操作に関しては関連研究で紹介した、Leap motion SDK によるジェスチャー入力方法を利用した。

2.2 両手の役割

本章では両手を使う際の、左手と右手の役割を説明する。左手の役割は、Leap motion SDK のジェスチャー操作を使う。それによって文字の削除等補助的な入力を行う。右手の役割は、グラフィティ文字入力という操作を行いつつ文字入力を行う。本研究において両手を使う理由は、バッティングの対策として右手・左手にそれぞれ役割を持たせた。

2.3 ジェスチャー操作

ジェスチャー操作は、Leap motion SDK に予め提供されている 4 つのジェスチャー関数を用いて入力をする。図 2.1 は SDK に用意されている、ジェスチャー例の画像である。

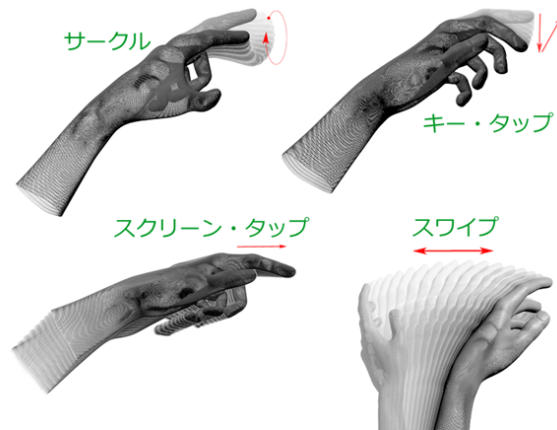


図 2.1 SDK ジェスチャー例

[文献 [5] の図 4 より転載]

手首から先を回すように動かすと「サークル」を認知する。回転の方向は指定していない。指を下に動かすと「キータップ」といった形でジェスチャーを認識する。このジェスチャーの中で

「スクリーンタップ」だけは他とは違い、奥行を認識するジェスチャーである。図 2.2 は、動作中の様子である。

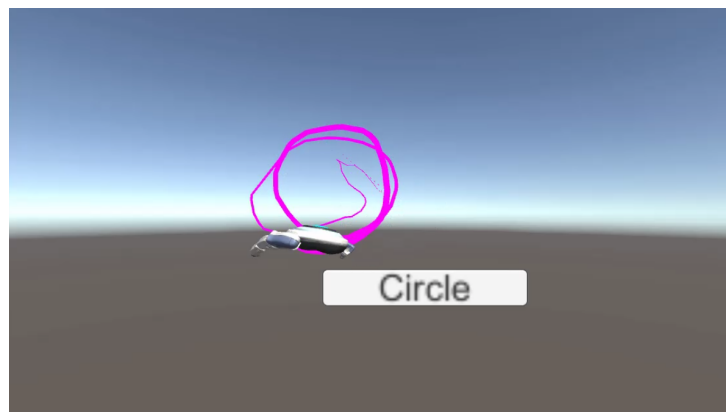


図 2.2 LeapSDK 動作中

2.4 グラフィティ文字入力操作

本節では「グラフィティ」と呼ばれる、手書き文字認識について述べる。グラフィティは一筆書き文字入力とも呼ばれており、元々は Palm 社（パーム社）が自社の PDA での文字入力方法 [15] として開発したものである。グラフィティ文字はアルファベットの形を簡略化したものであり、本来のアルファベット文字と比較して簡易な軌跡によって入力を可能としている。グラフィティ文字と似た手法で、片手操作で入力を行え、ジェスチャー入力を行おうとした研究例 [16] もある。既に Palm 社は自社 PDA の開発は行っていないが、現在でも Android 向けの文字入力方式 [17] として利用が可能である。図 2.3 は、グラフィティ文字の例である。

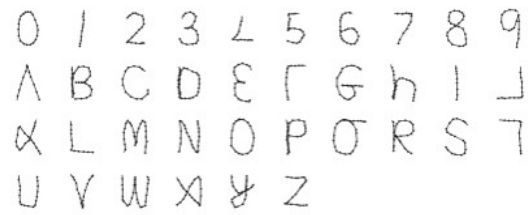


図 2.3 グラフィティ文字例

[文献 [10] の図 4 より転載]

グラフィティ文字は一般にアルファベットを崩した特徴的な形をしている。また数字も 4 は特殊な形をしている。このように、それぞれのアルファベットや数字に対応した簡潔な動きを認識させる事により、文字を入力する事が出来る。通常の文字と違い、グラフィティ文字はどの文字も全て一筆書きで入力 that 完結するという特徴がある。

2.5 グラフィティ文字入力の方法

入力方法として入力を可能にする準備が必要である為、図 2.4 に入力が可能になる手の形を示す。この正しい形をしなければ文字入力が行う事が出来ない。

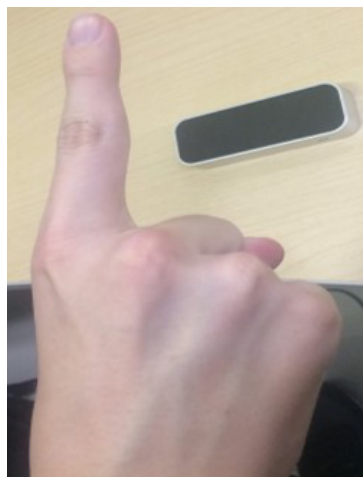


図 2.4 正しい手の形

親指の角度を検知している [18] ため、指定した角度にならないと入力する事は不可能である。

この形をすることで、人差し指の先にパーティクルが出現する。図 2.5 は文字が書ける準備が整った様子を表示している画像である。

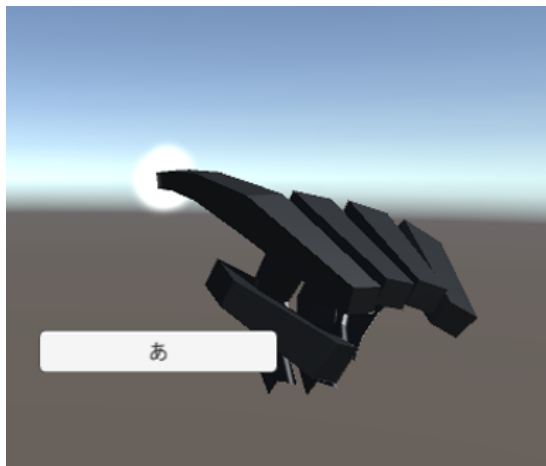


図 2.5 認識可能状態

検証する際の入力文字は、現在 4 つのグラフィティ文字によって入力出来る。下にある 4 つの図を本研究におけるグラフィティ文字と定義する。本研究の検証では特定の動きをした場合こちらで設定したひらがなを表示することで、ユーザーが入力が出来たかどうかを判断する。グラフィティ文字、仮に A を図 2.6 に示す。入力を行うユーザーが人差し指を右に動かしてから上へ動かした場合、グラフィティ文字 A と合致してひらがなの「あ」を表示する。ひらがなが表示されれば入力が成功したという事になる。グラフィティ文字 B は図 2.7 に示す。文字 A とは逆で、ユーザーが人差し指を左に動かしてから上へと動かすことで、グラフィティ文字 B と合致してひらがなの「い」を表示する。グラフィティ文字 C は図 2.8 に示す。ユーザーは人差し指を右に動かしてから下へ動かす。図 2.3 中で類似するのは T の形である。グラフィティ文字 C 合致した場合ひらがなの「う」を表示する。グラフィティ文字 D は図 2.9 に示す。ユーザーは人差し指を下に降ろしてから上に指を振る。図 2.3 中で類似するのは V の形である。グラフィティ文字 D と合致した場合ひらがなの「え」を表示する。上記が検証する際の現在考えている 4 つのグラフィティ文字である。



図 2.6 グラフィティ文字 A

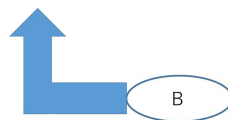


図 2.7 グラフィティ文字 B

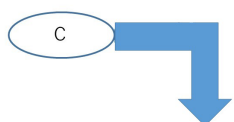


図 2.8 グラフィティ文字 C

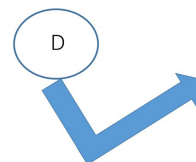


図 2.9 グラフィティ文字 D

検証方法として、文字を書くために指を動かした時の操作とプログラミングで決めたグラフィティ文字のデータを比較して、どれくらい近いかという事を示さなければならない。その為、本研究では定義したグラフィティ形状と指の操作の一致度を測定する為にハウスドルフ距離 [19][20] を用いて、計測を行った。以下の数式 (2.1) は、ハウスドルフ距離を示した数式である。

$$h(A, B) = \max_{a \in A} \{ \min_{b \in B} \{ d(a, b) \} \} \quad (2.1)$$

図 2.10 はハウスドルフ距離の例を示したものである。

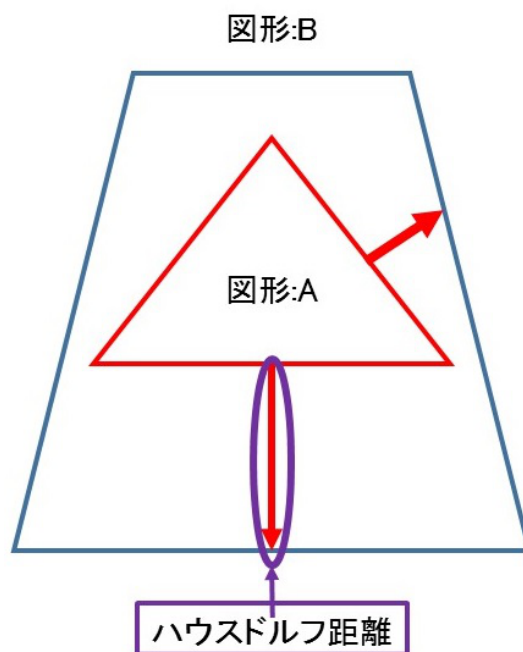


図 2.10 ハウスドルフ距離の例

ハウスドルフ距離とは、任意の図形 A と図形 B の辺から 2 つの図形の形状間の類似度を測定する指標として用いる。図形 A の辺から図形 B の辺までの最短距離を求め、その中から一番値が大きい最短距離を求め、その求めた距離がハウスドルフ距離となる。形状間が大幅に異なる物でも若干しか変化していなくても類似度を検出する事が可能である。図 2.10 を数式 (2.1) に当てはめると、 h はハウスドルフ距離を示し、 A は集合 A(図形 A の辺) を示し、 B は集合 B(図形 B の辺) を示している。 $d(a, b)$ は図形 A の辺から図形 B の辺までの距離を示している。

本研究では、図形の辺を点群に置き換えてその点群から近似的にハウスドルフ距離を求める手法を、入力文字推定の手法とした。求める図 2.11 に本研究におけるハウスドルフ距離の求め方を示す。

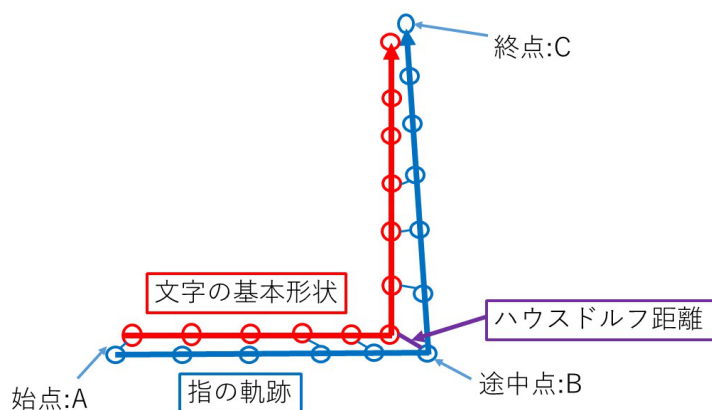


図 2.11 ハウスドルフ距離の求め方 (グラフィティ文字 A の場合)

図 2.11 に示してある赤い線は文字の基本形状である。青い線はユーザーが人差し指を使って動かし時の軌跡を描いている。まずユーザーが指で軌跡を描く。軌跡には一定間隔で点群を配置する。人差し指を動かし始めた始点から曲がる所を決める途中点と指を止めるまでの終点を記録する。終点を記録した後、赤い線で示している基本形状を読み込む。赤色の線上にも一定間隔で点群を配置する。そしてそれぞれの点群同士を配列を使い照らし合わせ、ハウスドルフ距離を求める。そうした上で求められた値から類似度が近いグラフィティ文字の形状と合致し文字を表示する。このハウスドルフ距離が求められれば形を読み取れることが可能であり、上記にあるグラフィティ文字を認識させる事も出来る。指の動きを認知する事で形が合致すれば文字入力が行えると言える。本研究では指による軌跡を入力後、基本形状と軌跡の始点を同一とし、基本形状の縦幅と横幅を軌跡と同一となるよう拡大補正を施し、ハウスドルフ距離を求めた。

2.6 両手を使った入力方法の説明

本章では両手を使った入力方法について述べる。図 2.12 は操作方法を使用した流れを示した画像である。これを元に説明を行う。

両手を使った操作の流れ

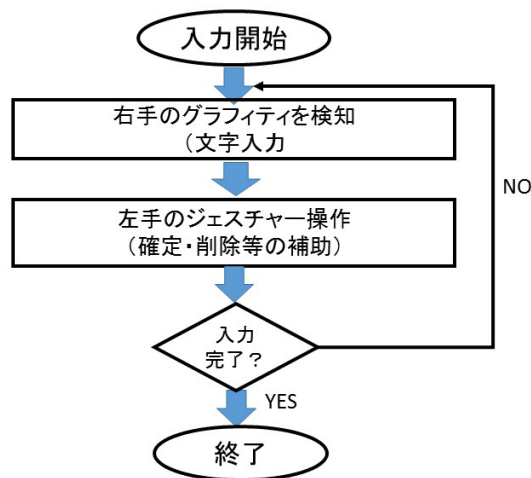


図 2.12 フローチャート

初めに右手によるグラフィティ文字入力を認識させ文字を入力していく。誤入力をした場合、左手によるジェスチャー操作を行い文字の削除といった補助的な動作を行う。右手で入力をし、左手で補助を行い入力を完了する。本研究では、右手と左手を交互に入れ替える入力方法を指標とする。何故交互に入れ替える必要があるのかと言うと、両手を同時に認識した状態で入力を行うとグラフィティ文字入力操作と SDK のジェスチャー操作がバッティングし、誤認識を引き起こしてしまった為である。また、片手で 2 つの操作それぞれを使う事は出来ないのか、という先行研究の問題と同じく 2 つの操作がバッティングし誤認識を引き起こしてしまった。その為、本手法では上記の右左を交互に入れ替える入力方法を両手を使った文字入力方法であると定義した。表示している手が右手であるか左手であるかの判断は、人差し指に対して小指の位置が x 線上で負の値だった場合左手とし、正の値だった場合右手と判断している。以上の手法を使い文字入力操作の検証を行った。

第 3 章

検証内容と結果

3.1 検証内容

本手法を実装し、20 代大学生 4 名の被験者による検証を行った。被験者は全員 Leap motion 未経験者である。検証の際には、手本となる動作を見せた後、実際にジェスチャ操作とグラフィティ入力を行い、検証を開始した。これを踏まえたうえで実行結果を記していく。それぞれの操作、検出結果を述べ、認識に要した時間、認識の精度を測定する。最後に、本手法における両手を使った文字入力方法に問題がないかの検証を行った。

3.2 ジェスチャー操作の検証方法と結果

1. 第 1 の検証として 4 種類のジェスチャー認識を初見で行ってもらいその最初の一回、その時に文字が表示されるまでにかかった時間を測定した。A~D は被験者を表す。文字がしっかりと表示されるまで、ジェスチャーは繰り返してもらった。表 3.1 は、認識時間の結果である。

表 3.1 ジェスチャー認識時間 (秒)

	A	B	C	D	平均
サークル	1.50	2.34	1.43	2.02	1.82
スワイプ	1.53	1.55	1.14	1.44	1.41
キータップ	7.32	6.31	10.90	6.13	7.66
スクリーンタップ	14.60	15.50	16.88	16.28	15.81

ジェスチャーの認識結果は、4人全員が入力に成功した。ただ、それぞれの動作を行っている際、誤認識が多いという現象が起こった。その原因は後に記す。表を見ると、サークル、スワイプは比較的認識しやすく、2秒程で全員が認識に成功した。しかし、キータップとスクリーンタップは、前者に比べ、時間を要していることが分かる。特にスクリーンタップは、全員が10秒以上、認識時間が経過してしまい、確実性が低い結果となった。

- 第2の検証としてジェスチャーの検出が出来たら、20回程同じ動作を繰り返してもらい、正しい動作が何回成功したか、認識の成功率を測った。その結果を表に示す。この検証は、10回1セットとして2回に分けて検証を行った。表3.2は、前半10回の成功率を測ったものである。

表 3.2 ジェスチャー操作の成功率 (前半 10 回)

	A	B	C	D	平均
サークル	8	10	9	10	9(90%)
スワイプ	4	7	8	10	7(70%)
キータップ	6	10	7	3	7(70%)
スクリーンタップ	0	0	10	8	5(50%)

検証1と同じく、サークルの動作は成功率が高い。スワイプ、キータップも、コツを掴んだためか、成功率は高いと言えるだろう。逆にスクリーンタップは成功率が低い結果となっており、A、Bは一度も認識に成功する事は、ここでは出来なかった。次に表3.3に、後半10回の成功率を記した。

表 3.3 ジェスチャー操作の成功率 (後半 10 回)

	A	B	C	D	平均
サークル	10	10	10	9	10(100 %)
スワイプ	8	10	10	10	10(100 %)
キータップ	6	9	9	6	8(80 %)
スクリーンタップ	1	8	10	10	7(70 %)

後半の結果は、成功率が上昇しているのが分かる。

最後に 20 回、ジェスチャー操作の成功率を測った結果を表 3.4 に記す。

表 3.4 ジェスチャー操作の成功率 (20 回)

	A	B	C	D	平均
サークル	18	20	19	19	19(95 %)
スワイプ	12	17	18	20	17(85 %)
キータップ	12	19	16	9	14(70 %)
スクリーンタップ	1	8	20	18	12(60 %)

20 回の試行により、スクリーンタップの検出率は高い被験者と低い被験者の双方が見られ、振れ幅が大きい結果が出た。スクリーンタップを 20 回、認識している被験者の動きを観察したところ、なるべく画面から平行に腕を構えて、前に腕をつき出すと、認識が上手く行く傾向が見られた。指をそろえながら動作を行うと、ジェスチャーの認識がしやすいという傾向がある事も分かった。

3.2.1 ジェスチャー操作、検証の分析

検証の分析として、サークルとスワイプの認識率が高く、キータップとスクリーンタップは、初見では認識し辛いという事が分かった。他のジェスチャー動作とバッティングしているのが原因と考える。繰り返してもらえると、ジェスチャーの認識はよくなり、難しいスクリーンタップも、コツをつかめば入力が可能という結果が出た。検証の結果から、文字の削除をする補助的なシステ

ムを行う際には、サークルを使用する。

3.3 グラフィティ文字入力の認識検証と結果

続いてグラフィティ文字入力の検証を行った。第2章の方で示した4つの文字入力を参照しつつ、検証を行っていく。被験者は変えずに検証を行った。図3.1はグラフィティ文字入力をしている際の画像である。

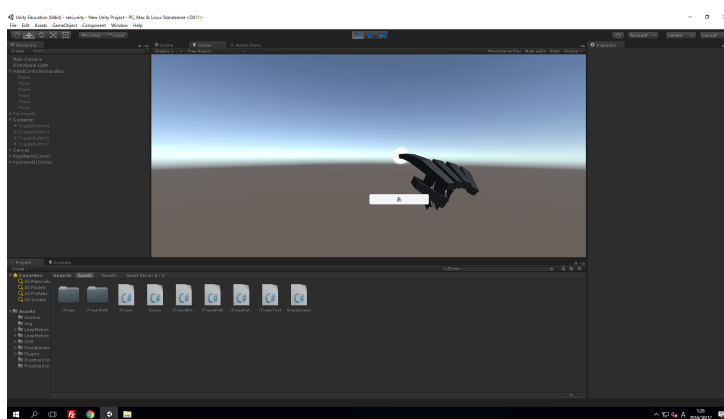


図 3.1 グラフィティ文字入力:検証中

1. 第1の検証としてそれぞれの文字入力動作を行ってもらい文字が表示されるまでにかかった最初の1回の時間を測定した。全ての動作を口頭で説明したうえで被験者達に入力検証を行ってもらった。表3.5は、その結果を記したものである。

表 3.5 グラフィティ文字認識時間 (秒)

	A	B	C	D	平均
グラフィティ文字 A	1.43	2.01	1.94	1.11	1.62
グラフィティ文字 B	1.26	1.81	1.92	1.67	1.67
グラフィティ文字 C	1.92	1.79	2.45	1.42	1.90
グラフィティ文字 D	1.72	1.73	1.58	2.30	1.83

全体的に入力にかかった時間はジェスチャー操作と比べてみると早いと言える。グラフィティ文字 C が一番時間がかかっている。

2. 第2の検証として、文字の検出が出来たらジェスチャー操作と同じく動作を20回繰り返してもらおう。それぞれの文字入力を正しい動作でしてもらい、何回成功したかを測定し認識の成功率を測った。表3.6にその結果を示す。

表 3.6 グラフィティ文字入力の成功率 (前半 10 回)

	A	B	C	D	平均
グラフィティ文字 A	8	7	7	8	8(80%)
グラフィティ文字 B	8	7	7	8	7(70%)
グラフィティ文字 C	7	6	6	6	6(60%)
グラフィティ文字 D	7	6	7	7	7(70%)

前半の結果を表に示した。認識の成功率は、グラフィティ文字 A と B は全体的に認識率が高い。グラフィティ文字 C は最も低い認識率を示した。表 3.7 は後半 10 回を測定した結果である。

表 3.7 グラフィティ文字入力の成功率 (後半 10 回)

	A	B	C	D	平均
グラフィティ文字 A	7	7	7	7	7(70%)
グラフィティ文字 B	8	7	6	7	7(70%)
グラフィティ文字 C	6	6	6	6	6(60%)
グラフィティ文字 D	8	7	7	7	7(70%)

後半の結果は、前半と比べてみても然程変わらない事がこの結果から分かる。最後にまとめたものを、表 3.8 に示す。

表 3.8 グラフィティ文字入力の成功率 (20 回)

	A	B	C	D	平均
グラフィティ文字 A	15	14	14	15	15(75%)
グラフィティ文字 B	16	14	13	15	15(75%)
グラフィティ文字 C	11	12	12	12	12(60%)
グラフィティ文字 D	15	13	14	14	14(70%)

成功率は成功と失敗が半々だという事が今回の結果から分かった。認識率は良好とは言えない結果となった。被験者の動作を観察したところ、入力動作を行い文字が表示されるまで若干のラグが存在した。また被験者の動作を観察したところ、余りにも早く指を振ると認識が出来なかった傾向が見られた。なるべく自然な動作で行うと認識がしやすくなる傾向がある事も分かった。

3. 第3の検証は4つの文字入力方法を続けてやってもらった。個別ではなく連続で「あいうえ」全ての文字が出るまでにかかった時間を検証した。どれくらいの時間で4つの文字全てが表示されたか、表3.9に結果を示す。

表 3.9 4つのグラフィティ文字が表示された時間

	A	B	C	D	平均
グラフィティ文字 A~D	8.01	11.56	12.94	11.01	10.88

スムーズに入力できて10秒という結果を示した。口頭意見で全員から言われた意見は「たまに違う文字が出て来る」「グラフィティ文字Cの認識率が悪い」という言葉が多かった。またバッティングしている傾向が見られ、入力時のラグのせいで文字が入力出来ているのかいないのか分からない事も多かった。検証中ではグラフィティ文字AとBが一番認識率が高かったがその分誤認識も多かった。グラフィティ文字Cを入力しようとする、グラフィティ文字Aとグラフィティ文字Dが認識される事が多く見受けられた。

3.3.1 グラフィティ文字入力操作、検証の分析

グラフィティ文字入力操作の検証分析として、認識の精度は良好とは言えない。認識出来る種類を4つのパターンで分けてみたが、成功と失敗のバラツキがあり実用的とは現段階では言えない。被験者からの口頭意見は「元となる文字を書く時とは全く関係の無い動きをして、文字が表示されるのは違和感が大きい」というのがあった。今回行ってもらった4パターンは右から上

や、V字型といった感じで指を移動させなければならないのだが、「常に手が浮いているので疲れる」「書いている実感があまり無い為不安になる」という意見ももらった。例え入力出来たとしても動作に関しては工夫がもっと必要である事に気付かされた。認識率についてはある程度の慣れとコツが必要であると今回の検証から分かった。比較的認識率が高かった被験者からは「緩やかに動かさず角ばってキビキビ動かした方が認識しやすい」という口頭意見を貰った。

3.4 両手を使用した時の認識検証と結果

続いて、両手を使った際の入力の検証を行った。それぞれの操作がバッティングしないか、認識が上手く行くかを検証した。被験者は変えずに検証を行った。図 3.2、3.3 は両手を使用し、検証を行っている際の画像である。



図 3.2 左手:ジェスチャー操作



図 3.3 右手:グラフィティ文字入力操作

1. 検証として、文字入力動作を行い 4 文字以上表示した後、ジェスチャー操作の認識率が高かったサークルを使用し文字を消去してもらう。その流れを 5 回繰り返してもらった。グラフィティ文字入力操作中にジェスチャー操作とのバッティングが起きるか、文字が書き込めるか消せるか、右左それぞれがしっかりと役割を果たしているか、認識の成功率の検証を行った。右手で文字を書いている時に文字が消える、逆に左手で文字が描けた場合は誤認識とし失敗と記録する。グラフィティ文字入力操作単体の誤認識は、この検証では考

慮しないものとする。被験者は変えずに検証を行った。表 3.10 に結果を示す。

表 3.10 両手を使った入力の成功率

	A	B	C	D	平均
両手使用	5	5	5	5	5(100%)

3.4.1 両手を使用した認識、検証の分析

まず右手と左手はそれぞれバッティングする事もなく、スムーズに動作していた。目立った誤認識もなく成功率も悪くない為、両手を使った文字入力操作は可能であるという事が分かった。被験者からは「グラフィティ文字単体の誤認識以外は快適」「両手を常に出しているよりかは楽」「両手を常に出して入力を行いたい」という口頭意見を貰った。

第 4 章

終わりに

本研究では Leap motion デバイスを用い、ジェスチャー認識とグラフィティ入力を使用した入力手法を提案した。Leap motion によるジェスチャー認識は高い実用性を持つと考えるが本研究における現時点では、まだ機能は限定的な段階に留まっておりその機能を活かしているとは言い難い。

現時点の課題として両手の同時認識がある。本研究では同時ではなく左右片手ずつの操作認識と定義したが、口頭意見から両手の同時認識は課題点と判断した。ただ、現段階でも同時に表示させる事と、右左に区別する事は可能であり図 4.1 にその様子を示している。

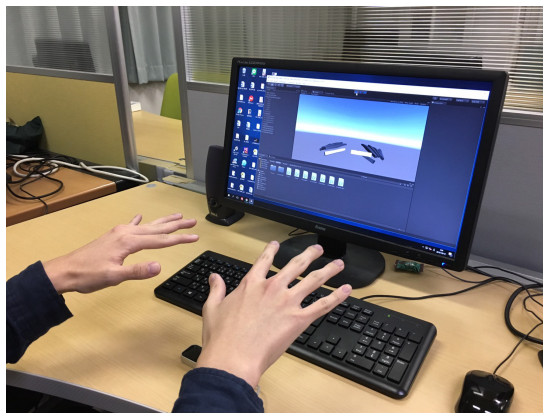


図 4.1 両手の認識

だが認識した後に動作を行うと、ジェスチャー操作とグラフィティ文字入力操作同士でバッティングが起こり、真面に文字入力が行えない状態に陥ってしまう。これは SDK が左右を判定しないジェスチャーだった為、改善が難しい点だった。また、片手だけ挙動がおかしくなってしまう問題がある。例えば掌が突然裏返る、指の一本だけが認識できず不自然な方向に折れ曲がったままの状態になる等が確認できた。この部分の認識については Leap motion の認識範囲の問題もある為、改善が難しい点だった。現段階の両手を使った文字入力も認識率は悪くないが、課題点と判断しこの部分の改善を目指していきたい。

また現状のグラフィティ手法によって、入力できる文字はかなり限られている。多くの文字入力を可能としないためバリエーションを増やしていく必要がある。だが現時点でのハウズドルフ距離を用いた手法では文字のバリエーションを増やしても、入力する文字のデザインにこだわっても入力がし辛く、誤入力の恐れもある為実用的な入力方法であるとは言い難い。ジェスチャー操作によってグラフィティ文字自体の誤入力を防ぐ事は出来ないため、誤入力したら消してもう一度書いてもらう必要がある。グラフィティ文字を認識する上で問題があると言えるだろう。また、文字の連続入力には課題が残る結果となった。一文字一文字の入力ではなく動作を連続で素早く行うと誤認識が多い事が本研究では判明された。この部分の改善も目指していきたい。ただグラフィティ文字は各文字の軌跡を覚える必要があるのだが、すぐに覚えられるような特徴的な一筆書き文字入力の形を作ることは現時点で可能である。また認識自体については 70 % 付近という認識率が検証結果から分かる。一文字一文字入力が完了してから動作を行えば、連続した文字入力でも誤入力の恐れは減っている事が分かった。グラフィティ文字入力は入力方法として向いていない手法ではない。本研究ではそれが分かり、認識度の向上や入力方法の見直しなど、本手法の完成度を高め、実用的な入力手法の提供を目指していきたい。

なお本論文は、第 14 回デジタルコンテンツクリエーション [21] にて「Leap motion を使用した文字入力方法の提案」という題目で、九州大学医学部百年講堂にて講演を行った。

謝辞

本研究を進めるにあたり、ご指導を頂いた卒業論文指導教員の渡辺先生、三上先生、阿部先生に深謝いたします。また、検証に協力してくれた方々、日常の議論を通じて多くの知識や示唆を頂いたゲームイノベーション・ゲームサイエンス研究室の皆様感謝いたします。

参考文献

- [1] ソフトウェアキーボードとは. <http://www.boy.co.jp/kojin/benri/myd/security/security03.html>. 参照 2016-12-19.
- [2] 久楽忠昭, 大西克実, 中野秀男ほか. 上肢障がい者向け入力支援における研究. 平成 22 年度 情報処理学会関西支部 支部大会, Vol. 8, No. 1, pp. 1–15, 2011.
- [3] 安井政人, 島村和典. 情報弱者用操作デバイスのためのソフトウェアキー入力の研究. Master's thesis.
- [4] 藤井祐介, 竹沢恵, 真田博文, 渡辺一央. 空中手書き文字入力システムの構築に関する一考察. 研究報告モバイルコンピューティングとユビキタス通信 (MBL), Vol. 2009, No. 6, pp. 1–4, 2009.
- [5] 細野敬太, 笹倉万里子, 田邊浩亨. Leap motion を用いたジェスチャ操作による文字入力方法の提案. 人工知能学会全国大会論文集, Vol. 28, pp. 1–4, 2014. 参照 2016-12-19.
- [6] Leap motion - official site. <https://www.leapmotion.com/>. 参照 2016-12-19.
- [7] 中村薫. Leap motion プログラミングガイド [改訂版], 第 No.1 巻. 2015.
- [8] Sharad Vikram, Lei Li, and Stuart Russell. Handwriting and gestures in the air, recognizing on the fly. In *Proceedings of the CHI*, Vol. 13, pp. 1179–1184, 2013.

- [9] 川上翔太, 篠原光, 近森友輔. Leap motion controller を用いた手の認識. Master's thesis, 2013.
- [10] 保呂毅, 稲葉雅幸. 複数カメラを用いた手書き文字認識システム. 第 14 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, 2006. 参照 2016-12-19.
- [11] 星野元気, 木村広. Leapmotioncontroller による空中仮想キーボード airkey の開発. Master's thesis, 2014.
- [12] Unity3d - official site. <http://japan.unity3d.com/>. 参照 2016-12-19.
- [13] 高星賢二, 山口武彦, 置塩正祐, 平山理恵, 原田哲也. 数式の抽象概念空間への身体的インタラクションの適用. Master's thesis, 2016.
- [14] 岩屋亮明, 澤田秀之. Vr エンタテインメントに向けたエア楽器演奏システム. インタラクション 2014 論文集, pp. 587–592, 2014.
- [15] うぐいすの鳴く季節に ～徒然なる日々～ かつて palm pilot というデバイスがあった. <http://shadowman.seesaa.net/article/406175333.html>, 2014. 参照 2016-12-19.
- [16] Kentaro Go, Yuichiro Kinoshita, and Shoichi Watanabe. Designing edgewrite versions for japanese text entry. In *Cyberworlds (CW), 2013 International Conference on*, pp. 120–123. IEEE, 2013.
- [17] Android 用一筆書き文字入力「graffiti pro for android」が無料化かつ高解像度端末対応に. <http://hitoriblog.com/?p=35039>, 2016. 参照 2016-12-19.
- [18] Leap sdk で指を検出してみよう (tracking hands, fingers, and tools) . <http://www.buildinsider.net/small/leapmotioncs/002>, 2015. 参照 2016-12-19.
- [19] 大人になってからの再学習 ハウスドルフ距離. <http://d.hatena.ne.jp/Zellij/20111206/p1>, 2011. 参照 2016-12-19.
- [20] 前瀬啓材, 原田祥吾, 呉海元, 和田俊和ほか. ハウスドルフ距離による近赤外線画像からの夜間

歩行者検出. 画像の認識・理解シンポジウム (MIRU2011) 論文集, Vol. 2011, pp. 703–709, 2011.

[21] 大西未来, 渡辺大地ほか. Leap motion を使用した文字入力方法の提案. 研究報告デジタルコンテンツクリエイション (DCC), Vol. 2016, No. 8, pp. 1–4, 2016.