

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

A proposal of the character input method by gesture operation using Leap Motion

細野 敬太*1

Keita Hosono

笹倉 万里子*2

Mariko Sasakura

田邊 浩亨*3

Hiroyuki Tanabe

川上 武志*3

Takeshi Kawakami

*1岡山大学工学部情報工学科

DEPARTMENT OF INFORMATION TECHNOLOGY FACULTY OF ENGINEERING, OKAYAMA UNIVERSITY

*2岡山大学大学院自然科学研究科

GRADUATE SCHOOL OF NATURAL SCIENCE AND TECHNOLOGY OKAYAMA UNIVERSITY

*3アイピーシステム株式会社

I.P. SYSTEM INC.

We propose a Japanese input method by hand operation using a Leap Motion that is a 3D motion sensor device. This paper describes a system that let a user to enter a hiragana character by the movement of fingers detected by a Leap Motion. The system is designed to input a hiragana by the combination of right hand for consonants and left hand for vowels. The result of a usability test of the system is not what we are satisfied by because the system makes a lot of different characters than the one that the users intend. We discuss the cause of input errors from the result of velocity graphs of fingers.

1. はじめに

1980年代, パソコンはキーボードを用いたコマンド入力で行う操作するキャラクタユーザインタフェースが主体であった. そこからパソコンは進化し, アイコンや画像を多用しマウスのようなポインティングデバイスによって操作するグラフィカルユーザインタフェースに変化してきた. 近年では, 機器の多機能化, 高性能化により, ジェスチャや音声などで機器を操作することが可能となった. そのため, マイクロソフト社の Kinect やスマートフォンなどに用いられるタッチパネルのように, 人間の自然な振る舞いや動作により機器と対話することができるナチュラルユーザインタフェースが普及してきた [1][2].

ナチュラルユーザインタフェースを実現するために, モーションセンサデバイスが用いられているものがある. モーションセンサデバイスの最大の特徴は, 身体の動きやジェスチャで非接触操作を実現できる点である.

2012年, 空中での手や指の動きを認識しコンピュータを操作することができる3次元モーションセンサデバイス Leap Motion(図1)が開発された. Fran Weichertらにより, Leap Motionコントローラの精度と正確さの分析についての論文が発表されている [3]. この論文によると, Leap Motionにより検知される指や道具など座標の変化の精度は0.2ミリメートルと確認された. 空中でのジェスチャによる文字入力, ウェブカメラを利用した空中手書き文字入力 [4] や両手の動きを利用した文字入力 [5] などの研究がある.

本研究では, 検知される両手の指の本数や, 動作の違いにより文字入力を行うシステムを設計し実装する. 指を基準値よりも早く動かすことで入力のための動作を行ったと判断し, 両手の指の動作角度の違いにより, ひらがな文字を入力する方法を提案する. また, 句読点などの記号入力は Leap Motion 用の開発ツールセットである Leap Motion SDK によりあらかじめ提供されている4つのジェスチャ関数を用いて入力する.

連絡先: 細野 敬太, 岡山大学工学部情報工学科,
k.hosono@momo.cs.okayama-u.ac.jp



図 1: Leap Motion Controller*1

右手から指が5本検知されると Enter, 左手から5本検知されると Delete 入力の操作をそれぞれ行う. 実装したシステムで入力に使用したジェスチャが文字入力に適しているかの評価を行う.

2. Leap Motion により得られる情報

Leap Motion では, 1秒間で200フレームの処理が行われている. Leap Motion SDK を用いて取得できる情報のうち, 本システムに使用するものを以下に記す.

検知された手の情報

指の数, 位置座標, 速度, 最も左にある手の情報, 最も右にある手の情報, 手の ID

検知された指の情報

座標, 速度, 各指の ID

検知されたジェスチャの情報

ジェスチャの種類, ジェスチャの状態

3. ジェスチャによる文字入力

3.1 入力可能文字

本システムでは以下に記す文字や記号について入力できるように設計した.

*1 <http://www.leapmotion.com> より引用



図 2: 文字入力初期状態

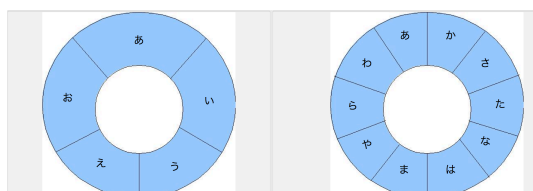


図 3: 文字入力画面

- 濁音・半濁音を含まない，ひらがな文字全て
- Enter・Delete キー
- 句読点，スペース，ハイフン

これらの文字入力を可能とするために，右手と左手で別々の動作を取得し区別することができるようにする．

3.2 文字入力方法の設計

文字入力を行う初期状態を，図 2 のように Leap Motion に対して両方の手をかざした状態とする．検知された手の指の数やジェスチャの種類によってユーザは文字を入力することができる．入力文字が確定すると，対応する文字のキーコードを入力する処理をプログラムで行う．

ひらがな文字の入力

ひらがな文字の入力は，図 2 の状態から Leap Motion が両手の指を 1 本ずつ検知した時に行うことができる．この状態で右手や左手の指を動かし，それぞれの指の動作方向の違いによって文字を入力する．

ひらがな文字を入力する時，図 3 の画面が表示される．右側の円は右手に，左側の円は左手にそれぞれ対応している．指先の初期位置を円の中心と考え，入力方向に右手と左手をそれぞれ動作させると文字が入力される．右手では子音を，左手では母音をそれぞれ選択する．

例えば，右手を“ま”，左手を“う”の方向にそれぞれ動かすと，ひらがなの“む”が入力される．このとき，右手と左手の入力順は影響されず，両方の入力が確定すると文字が入力される．“ん”はひらがな文字入力を行う状態で，図 4 のスクリーンタップを行うと入力できる．

句読点，スペース，ハイフンの入力

句読点，スペース，ハイフンを入力するには，図 2 の状態から Leap Motion が片方の手のみ指を検知した状態でジェスチャ

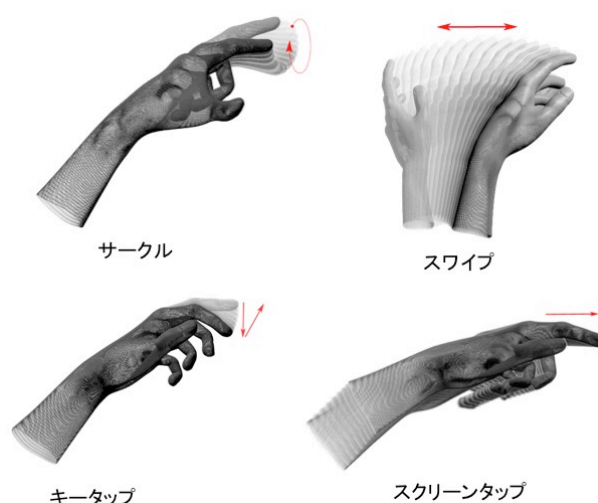


図 4: 句読点，スペース，ハイフン入力のジェスチャ*2

を行う．この時に行うジェスチャは図 4 に示す Leap Motion SDK により提供されている 4 種類のジェスチャ関数を用いる．

サークル 「。(読点)」の入力を行う．

スワイプ 「(ハイフン)」の入力を行う．

キータップ 「(スペース)」の入力を行う．

スクリーンタップ 「,(句点)」の入力を行う．

Enter，Delete キーの入力

Enter の入力は，右手を開いて Leap Motion が右手の指を 5 本検知すると行われる．左手を開いて Leap Motion が左手の指を 5 本検知すると Delete の入力処理を行う．

4. アルゴリズム

本システム全体のアルゴリズムを以下に示す．Leap Motion はフレーム処理であるため，1 フレームの処理に関する処理を表している．

全体のアルゴリズム

```

初期処理
エラー処理
if (手が 0 本でない) //手を検知
    if(手が 2 本) //手を 2 本以上検知
        右手と左手取得
        if(指が両手にある) //両手の指を検知
            ひらがな入力処理
        else
            記号入力処理
    
```

ひらがな入力処理

```

if(右手の指が 5 本)
    Enter キー入力処理;
    
```

*2 http://developer.leapmotion.com/documentation/objc/devguide/Leap_Gestures.html より引用

```

else
  右手の指を取得;
  指の移動角度取得処理;
if(左手の指が5本)
  Delete キー入力処理;
else
  左手の指を取得;
  指の移動角度取得処理;
if(入力がジェスチャ関数のスクリーンタップなら)
  “ん”の入力処理;
if(両手の入力が確定しているなら)
  文字入力処理;
    
```

記号入力処理

```

if(右手の指は5本)
  Enter 入力処理;
else
  ジェスチャ取得処理;
  記号入力処理;
    
```

Leap Motion SDK のみでは右手と左手がそれぞれ行った動作を区別することができないため、それらを区別できるように実装した。

5. ユーザによる評価実験

5.1 評価方法

本システムを実際に使用してもらい、入力文字に対して違う入力を行った回数を Delete キーの入力回数により比較する。ひらがな入力は、右手は 10 種類の動作方向を 1 つずつ、左手は 5 種類の動作方向を 2 つずつ行えるように指定した。ひらがな 10 種類と“ん”、さらに記号 4 種類の 15 種類の文字入力の測定を行う。

5.2 評価環境

以下の 2 つの測定環境によって測定を行った。測定環境 1 を以下に示す。

コンピュータ

OS : Mac OS X 10.7.5, プロセッサ : 2.8GHz Intel Core i5, メモリ : 8G 1333MHz

被験者

20~50 代の社員 10 人。仕事でシステム開発をしている。

測定環境 2 を以下に示す。

コンピュータ

OS : Mac OS X 10.9.1, プロセッサ : 2.6GHz Intel Core i5, メモリ : 16G 1600MHz DDR3

被験者

20~30 代の男性 4 人。事務仕事や大学の研究等でパソコンを使用している。

5.3 評価結果

各文字に対して、Enter キーが押されることで確定するまでに入力された Delete キーの回数を被験者ごとにまとめた結果を表 1 に示す。測定環境 1 での被験者を A~J, 測定環境 2 での被験者を a~d と表す。評価結果の数字は、対象の文字が正しく入力され、Enter キーが入力されるまでに Delete キーが

表 1: 入力文字に対する Delete キーが入力された回数

	あ	き	す	て	の	ひ	む	よ	れ	わ	ん	ス ペ ー ス	。	、	—	合 計	平 均
A	8	12	38	0	2	2	13	11	27	48	18	0	10	10	4	203	13.5
B	27	7	9	7	33	12	5	8					3	4	4	119	10.8
C	0	2	1	9	7	4	1	1	1	1	1	2	8	5	6	49	3.3
E	19	15			19	1	1	4	27		2	5	0	14	6	113	9.4
F	3	18	17	3	1	6	1	4	3		9	0	10	9	17	101	7.2
G	3	25	43	17	26	9	23	72	22	28	0	44	2	48	8	370	24.7
H	8		17	10		5	6	2	31	2	2		24		19	126	11.5
a	7	17	37	5	10	1		1	18	4	2	13	7		8	130	10.0
b	4	0	5	13	3	0	6	1	13	9	0	3	2	0	1	60	4.0
c	7	13	6	5	5		6	5	66	50		8		26	30	227	18.9
平均	8.6	13.8	19.3	7.7	11.8	4.4	6.9	10.9	23.1	20.3	4.3	9.4	7.3	14.5	10.3		

入力された回数を示す。そのため、本評価においては、正答したもののみを分析の対象とする。被験者 I, J, d は評価を最後まで行うことができなかったため、また、被験者 D は左利きであり、右利きの人と測定に差が生まれたため本評価においての比較対象から外すこととする。

評価実験の結果、文字の入力回数に対して非常に多く Delete キーが入力されたため、本システムに用いたジェスチャ操作は文字入力に適したものではないと言える。指を素早く動かさないと文字入力にならないこと、右手の入力を細かくしすぎたこと、入力後指を元の位置に戻す動作が文字入力動作として検知されたこと、Enter, Delete 入力の指の動きを文字入力のための動作と捉えて処理を行ったことが原因として考えられる。

また、記号入力の際にひらがなが入力されること、Delete キーや Enter キー入力の操作を 1 回だけ行ったのに複数回入力されることなどの誤検知も多かった。手の角度により、手を開いていても指が検知されないなど、ユーザの行った動作とセンサにより検知した動作にズレがあったためだと考える。また、Leap Motion SDK に使用されているジェスチャ関数は何も変更せずに入力するように実装したが誤入力が多かった。そのため、素早いジェスチャ操作を単純に処理しただけでは、文字入力のような入力回数の多い処理を行うことは現実的ではないと考えることができる。

6. 指の動作スピード測定

6.1 評価方法

評価実験 1 の結果から、誤入力の原因を調べるために新たに実験を行った。ひらがな入力において、入力のために指を動作させた場合に検知される指の動作スピードの変化を測定する。本実験では、右手のみの動作スピードの変化を測定する。8 方向に対して入力を行い、指示された方向への入力が確定するまで入力を続けて行う。また、指の位置を元に戻す動作は、正しい入力が行われたか確認してから行った。入力のための動作と指の位置を元に戻すための動作を区別する基準値を 1000[mm/sec] とした。

6.2 評価環境

コンピュータ

OS : Mac OS X 10.7.5, プロセッサ : 2.8GHz Intel Core i5, メモリ : 8G 1333MHz

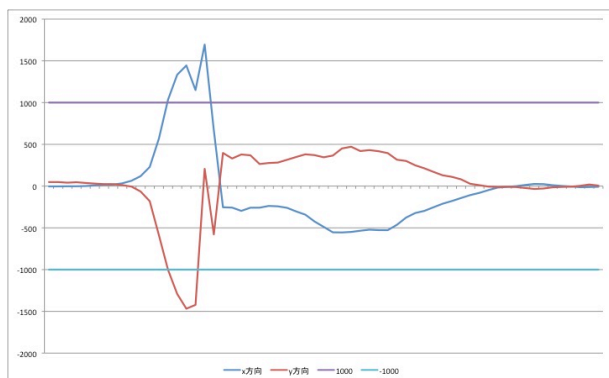


図 5: 右下を検知した波形

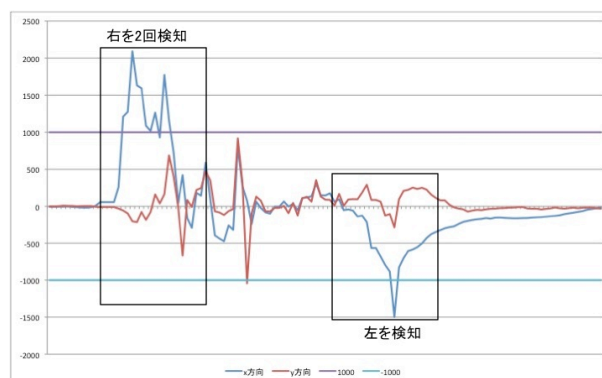


図 7: 右を 2 回, 左を 1 回検知した波形

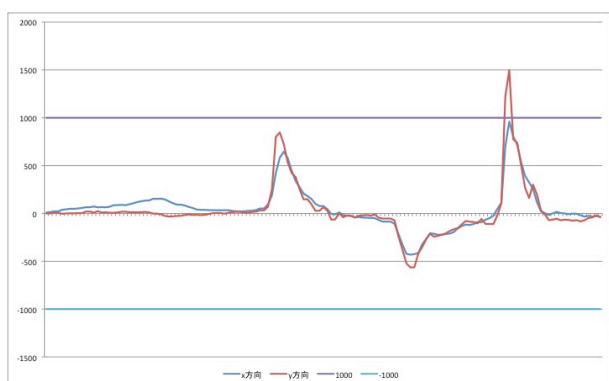


図 6: 右上を検知した波形

被験者

20 代の学生 8 人

6.3 評価結果

測定において、右下の動作を検知した時の波形を図 5 に、右上を検知した時の波形を図 6 に、右を 2 回、左を 1 回検知した時の波形を図 7 に示す。ここでは、青い波形を x 方向、赤い波形を y 方向の速度を表す。

図 5 では、x 方向の速さの値が正方向に、y 方向の値が負の方向に大きくなっている。また、入力動作を検知した後、入力方向とは逆方向への速さの値の変位が確認された。図 6 では、x 方向の速さの値が正方向に、y 方向の値が正の方向に大きくなっている。また、入力動作を検知する前、入力方向とは逆方向への速さの値の変位が確認された。図 7 では、x 方向の速さの値が正方向に大きくなっている。入力動作が 1 回行われたのに対し、2 回行われたと検知された。また、入力動作を検知した後、入力方向とは逆方向への速さの値の変位が確認された。

入力動作とは別の動作を検知した原因は、入力動作時に反対方向への指の動作スピードの値が得られたことであると分かった。図 7 のように、反対方向へ検出されたスピードの値が基準値を超えた場合、その時のわずかな指の座標変化を計算し、入力動作とは別の入力として上書きする。これは、図 5 のように動作後に現れる場合と、図 6 のように動作前に現れる場合がある。また、図 7 のように 1 回の入力動作中に 2 回基準値をまたぐスピード値の変化があった場合、2 回入力が行われたと検知され誤入力につながる。

7. おわりに

本研究では、3 次元モーションセンサデバイスである Leap Motion を用いて、両手の指を動かす、その動かす角度によりひらがなを入力する方法を開発した。右手では、ひらがなの子音を、左手ではひらがなの母音を入力する。また、Leap Motion SDK 含まれている 4 種類のジェスチャ関数を用いて、句点、読点、ハイフン、スペースの入力を行う。評価実験を行った結果、本研究で実装したジェスチャ操作における文字入力の方法はユーザにとって使いにくいものであることが分かった。

今後、アルゴリズムの改良、どんなユーザでもストレスを感じない入力方法の開発を進めたい。そのために、細かいユーザビリティ評価、センサとユーザの壁をなるべくなくす実装を心がけて開発を行いたい。

謝辞

評価実験に協力していただいた皆様に深く感謝いたします。

参考文献

- [1] モバイル UX を変えるための、インターフェースの現在/過去/未来: www.infobahn.co.jp/sphere/2906
- [2] 魔法の UI: <http://techon.nikkeibp.co.jp/article/HONSHI/20120426/215313>
- [3] Weichert, F., Bachmann, D., Rudak, B. and Fisseler, D.: Analysis of the Accuracy and Robustness of the Leap Motion Controller, Sensors, Vol.13, pp.6380-6393 (2013).
- [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, 情報処理学会研究報告, Vol.2009-MBL-50, No.6, pp.1-4 (2009).
- [5] 中村卓, 高橋伸, 田中二郎; "Hands-Popie: 両手の動きを利用した日本語入力", 日本ソフトウェア科学会 WISS2006, pp.151-152 (2006).