モーションセンサを用いた非接触でのフリック入力

2018SC091 田中瑞季
指導教員：大石泰章

1　はじめに

昨今，コロナ感染症の影響により人々は不特定多数が用いる機器への接触に抵抗感を持つようになっている．また，アフターコロナでは，非接触を実現しながら経済を成長させていく技術に注目が集まっている [1]．我々の生活において，街頭での商品検索や鉄道の券売機での行先の検索などタッチパネルでの文字入力を行う機会が多くある．そのような機会に非接触での文字入力ができれば，機器の使用に対する抵抗感をなくすことができると考えられる．

先行研究では，一方の手で子音の入力，もう一方の手での母音の入力をすることによって非接触の文字入力システム実現したものがある [2]．しかし，その場合，両手の動きをそれぞれ判断しなくてはいけない．

本研究では，フリック入力による非接触の文字入力を行うシステムを制作する．フリック入力は文字入力方法として多くの人になじみがあり，特殊な入力方法を覚える必要がなく，片手での文字入力を実現することができるため，非接触の文字入力法として受け入れやすいと思われる．ジェスチャーを識別する機能があらかじめ実装されており，特にスワイプを認識できる 3D モーションセンサである Leap Motion を使用する．

2　使用機器

本研究では，3D モーションセンサ Leap Motion（図 1）および，Leap Motion 専用開発ツールセット Leap Motion SDK を用いる．

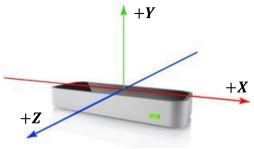


図 1: Leap Motion

Leap Motion は米 Leap Motion 社が販売している，手や指の検出に特化したセンサである．Leap Motion は 2 基の赤外線カメラと赤外線照射 LED から構成されている．開発ツールセット Leap Motion SDK は，Leap Motion のトラッキングデータに対する API を提供しているシステムである．Leap Motion SDK の関数のうち，本研究で使用するものを表 1 に示す．ただし，XYZ 座標は，図 1 に示すように，Leap Motion の LED ライトを正面に向けて置いたとき，左右方向に X 軸，上下方向に Y 軸，前後方向に Z 軸を取って定義する．

本研究ではスワイプの認識が重要であるので，X, Y, Z 軸それぞれに対するスワイプの認識範囲の測定を行った．

表 1: 使用した関数

関数名	関数の説明
Hand.getX()	手の中心の X 座標を取得
Hand.getY()	手の中心の Y 座標を取得
Finger.isExtended()	指が曲がっているかどうかを検出
Swipe.direction()	スワイプの方向を取得

結果は表 2 のとおりである．ただし，紙面の都合で Z 軸方向の測定結果は割愛する．表の結果から X 軸，Y 軸，Z 軸方向に対する十分にスワイプが認識できる範囲はそれぞれ，$-250<X<250$, $150 \leqq Y<650$, $-300<Z<300$ (単位は mm) と考えられる．以下で作成するシステムはこの範囲で行うようにする．

表 2: 50 回のスワイプ動作を行ったときの認識率

(a) X 軸方向の認識率

原点からの距離 (mm)		認識率 (%)
0 以上	50 未満	100
50	100	100
100	150	100
150	200	90
200	250	78
250	300	26

(b) Y 軸方向の認識率

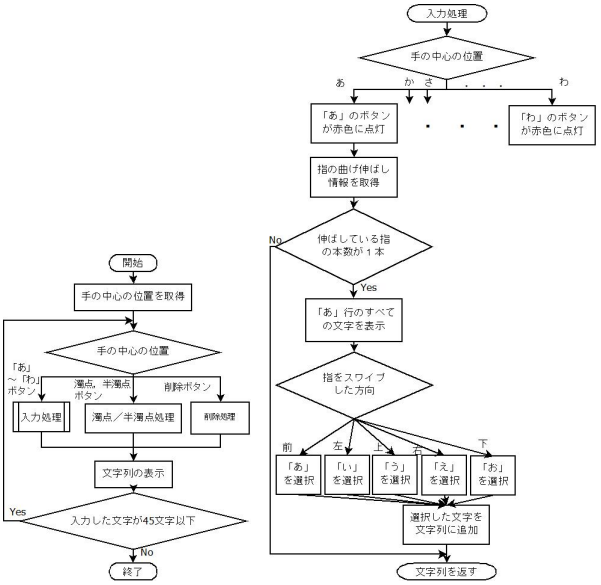
原点からの距離 (mm)		認識率 (%)
50 以上	150 未満	51
150	250	100
250	350	100
350	450	100
450	550	94
550	650	92
650	750	68
750	850	30

3　文字入力の手順

3.1　概要

本研究では図 3(a) のようなボタンが仮想的に空中にあるものとし，これを押したりスワイプしたりする動作を Leap Moton で読み取ることで，文字入力を実現する．現在の手の位置に対応するボタンがどれであるかや，そのボタンが選択されたかなどは，画面上に表示されるので，使用者はそれを見ながら操作できる．入力された文字は画面上に 45 文字まで保存され，45 文字目を入力されたところでシステムが終了する．

このシステムの動作フローを図 3(a) に示す．まず，手の中心の X 座標と Y 座標を表 1 の関数 Hand.getX() と Hand.getY() を用いて取得する．手の中心の位置がいずれかのボタンに対応する場合，対応するボタンに基づいて 3.2〜3.4 節で述べる処理を行う．結果として得られた文字列を表示し，文字列が 45 文字未満なら新たな文字入力を受け付け，45 文字のときは終了する．

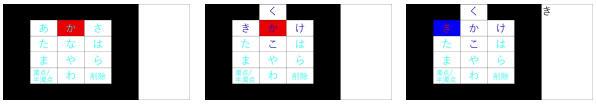


(a) システムの全体図　　(b) 入力処理

図 2: システムの構成

3.2　通常のボタンに対する処理

手の中心の位置が「あ」～「わ」のボタンに対応する場合の処理を，図 2(b) のフロー図に沿って説明する．はじめに，3.1 節で述べた方法で取得した手の中心の位置によって画面上の「あ」～「わ」のいずれかのボタンが赤色に変化する．次に，伸ばした指の本数が 1 本にすると，選択したボタンに属する全ての文字が画面上の表示される (図 3(b))．これは関数 Finger.isExtended() を利用して判定している．伸ばした指が 1 本のときに指をいずれかの方向にスワイプして文字を入力する．スワイプの方向は Swipe.direction() を用いて検出できる．スワイプして文字入力を行うと対応したボタンが青色で表示され，文字列に選択した文字が追加される (図 3(c))．そして，文字列を返してこの処理が終了する．



(a)　　(b)　　(c)

図 3: 入力処理の様子

3.3　「濁点／半濁点」のボタンに対する処理

手の中心の位置が「濁点／半濁点」のボタンに対応する場合の処理を説明する．この処理でも，3.2 節で述べた処理と同様に伸ばした指の本数が判定される．指が 1 本になったときに指を押し込むように動かすと画面上に表示されている文字列の末尾の文字に濁点，半濁点を付けたり，消したりが行われる（図 4(a)）．押したとき，「濁点／半濁点」ボタンは青色に表示される．このとき，は行の文字の場合はこのボタンを選択するたびに濁音，半濁音，清音の順に変化していく．このようにして，末尾を変更した文字列を返してこの処理が終了する．

3.4　「削除」のボタンに対する処理

手の中心の位置が「削除」のボタンに対応する場合の処理でも，3.2 節で述べた処理と同様に伸ばした指の本数が判定される．指が 1 本になったときに指を押し込むように動かすと画面上に表示されている文字列の末尾が削除される（図 4(b)）．押したとき，「削除」ボタンは青色に表示される．このようにして，末尾を削除した文字列を返してこの処理が終了する．

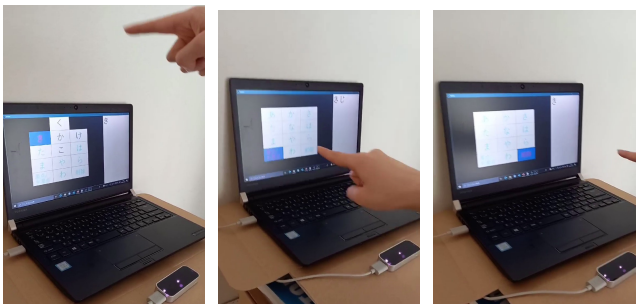
(a) 濁点／半濁点処理　　(b) 削除処理

図 4: 濁点／半濁点処理（左図）と削除処理（右図）の様子

4　文字入力システムの制作

前述で述べた文字入力の手順を Leap Motion を用いて実装した．制作したシステムによる文字入力の様子を図 5 に示す．

図 5(a) のように「か」のボタンの位置で指を 1 本にして左にスワイプすると「き」が入力される．また，「し」を入力した後に図 5(b) のように「濁点／半濁点」ボタンの位置で前に押し込むと「し」が「じ」に変換される．最後に図 5(c) のように「削除」ボタンで押し込みを行うと「じ」が削除される．



(a)　　(b)　　(c)

図 5: 文字の入力の様子

5　おわりに

本研究では，Leap Motion を用いてひらがなのフリック入力を非接触で実現することを試みた．Leap Motion SDK が認識できるジェスチャの 1 つであるスワイプを利用することによって非接触でもフリック入力と似た動きによって文字入力ができる．

今後の課題として，入力した文字から予測変換を出すシステムの実現が考えられる．また，本研究のシステムを使用した被験者から手を握るときの動作がスワイプとして認識されるなどの意見があった．

参考文献

[1]「アフターコロナ時代の新常識、「非接触」５大技術を分析」.『日経クロステック Active』, 2020 年 5 月 27 日, https://active.nikkeibp.co.jp/atcl/wp/b/20/05/21/00552/

[2] 井上賢人，梅澤猛，大澤範高，「両手の指と掌の接触を用いたかな文字入力手法の検討」,『第 81 回情報処理学会全国大会講演論文集』, 福岡, 2019 年 3 月, pp.85-86