

Wii リモコンから得た加速度によるオンライン文字認識方式

松永 啓佑*¹, 村上 仁己*², 小池 淳*³

Online Character Recognition Method of Acceleration by Wii Remote Controller

Keisuke MATSUNAGA *¹, Hitomi MURAKAMI *², Atsushi KOIKE *³

ABSTRACT : As for online character recognition by position coordinates of the input point using tablet, many studies are conducted than before. However, examinations using value of acceleration provided by acceleration sensor equipped with by Wii remote controller are not done very much. In this study, we propose a recognition method that the acceleration to be provided from Wii remote controller used about several sets of online recognition methods and examine effectiveness.

Keywords : Wii remote controller, Online Recognition

1. はじめに

タブレットを用いた入力点の位置座標によるオンライン文字認識は以前より数多くの研究が行われてきている[1][2]。しかし、Wiiリモコン等に搭載されている加速度センサーによって得られた加速度の値を利用した検討はあまりされていない[3][4]。本研究では、数式のオンライン認識方式について、Wiiリモコンから得られる加速度の利用した認識方式を提案し、有効性を検討する。

Wiiリモコン

Wiiは2006年末に発売された、任天堂の家庭ゲーム機である。Wiiリモコンは無線化された片手で持つ、モーションセンサーを主軸に置いたコントローラーである。指先ではなく全身の動作に注目したヒューマンインタフェースを標準搭載したWiiは家庭用ゲーム機において強いインパクトがあった[2]。

Wiiリモコンの無線機能はBluetooth接続であり、ホストアダプタがあればPCで使用することが可能。また、フリーツールのWiimoteLibを使用することで容易にプログラムを作成することが可能である。

加速度センサー

Wiiリモコンにはx,y,zの三軸方向に加速度センサーが実装されており、前後、左右、上下の加速度を検出が可能である。位置入力には加速度センサーでは検出出来ず、先端にあるポインターで赤外光を取得することによって検出する。

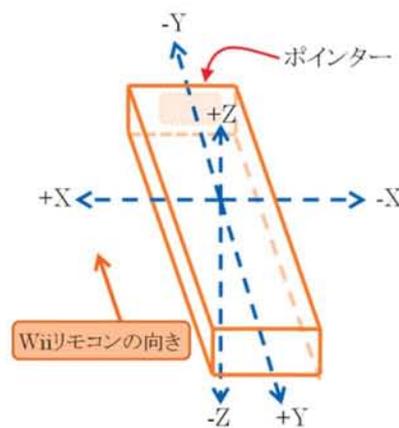


図1. Wiiリモコン

*¹ : 理工学研究科理工学専攻

*² : 理工学部情報科学科

*³ : 理工学部情報科学科 (koike@st.seikei.ac.jp)

2. 認識方法

Wiiリモコンを使用し、図2のようなウィンドウに数字や記号（以後、文字と略）を筆記する。ウィンドウの大きさは800×800である。

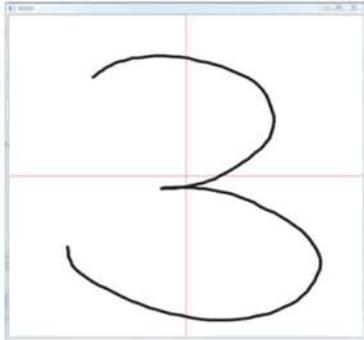


図2. 筆記用ウィンドウ

筆記と同時に図3のメインフォーム内にあるピクチャーボックスに表示される。図3ではそのピクチャーボックスは円で囲んだ数字の3のことである。文字が表示されたと同時に構成する代表点の座標値が表示される。これらの代表点の位置情報を基に後述の3つの方式を使用する。



図3. メインフォーム

2.1 代表点

本研究において代表点には位置座標、方向コード、加速度がそれぞれ設定される。1ストロークの長さを等間隔で区切った位置に8つ設置している。

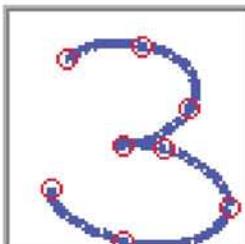


図4. メインフォームで表示された文字

文字認識は予め作成した辞書に登録された文字の代表点のデータと筆記した文字の代表点のデータを比較して行う。図5は対応する各代表点同士を線で繋いだものである。予め作成した辞書には代表点数や文字数、それぞれの代表点に対しての位置座標、方向コード、加速度が登録されている。

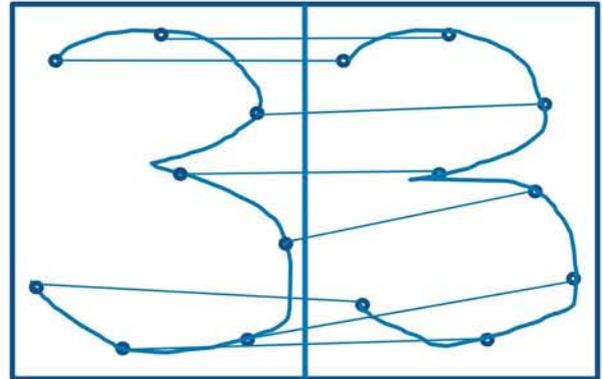


図5. 対応する各代表点

3. 既存の認識方式

方式1 位置座標系列を用いた認識方式

筆記した文字の代表点のx,y座標とその点に対応する登録済みの標準文字の代表点のx,y座標とのユークリッド距離を求める。全ての代表点で行い、その総和が最も小さくなる標準文字を見つけて認識結果とする方式である。この時、文字の重心が一致していない、大きさが異なる時には文字が正しい場合でも総和が大きくなり、正しい結果が出力されないことがある。

ユークリッド距離を図6で示す。代表点と代表点を結んでいる線は対応を示している。線で繋がれたそれぞれの代表点のxおよびy座標を比較する。

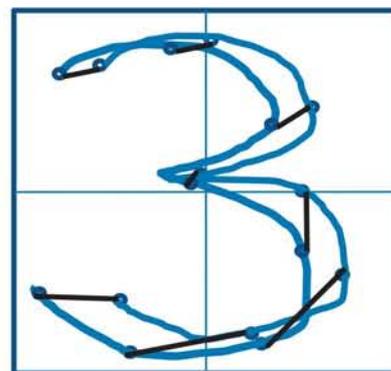


図6. ユークリッド距離

認識に使用する計算式は式①を参照。

$$D = \sum_{i=0}^{n-1} \{ (MX_i - TX_i)^2 + (MY_i - TY_i)^2 \} \quad \text{①}$$

Dは総和、nは代表点の数、MX、MYは登録済みの文字の各座標、TX、TYは認識したい文字の各座標である。

方式2 方向コード列を用いた認識方式

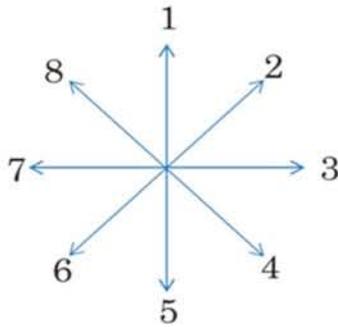


図7. 方向コード

代表点において図7のように上、右上、右、右下、下、左下、左、左上、8方向のいずれかを割り当てる。方向コードは代表点と次の代表点の座標の位置関係から求める。標準文字の代表点における方向コードと比較し、その差を求める。例えば、標準文字が上であり、筆記した文字が右に近い場合は3となる。全ての代表点で差を求め、合計差異が最も少ない標準文字を見付け出す方法である。この方法は、方式1に比べて文字の向きの変化に対する耐性がある。

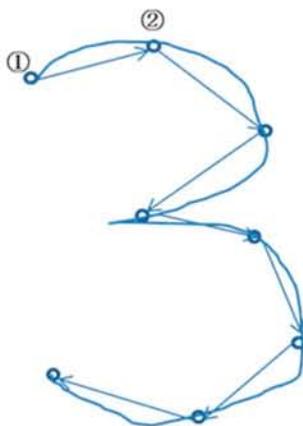


図8. 代表点間の角度

ここで、図8の①から②の方向を例にとって説明する。図8にある方向コードに①から②の角度を表したベクトルを当てはめると図9となる。2と3の間にあり、更に

3の近くにある。よって、代表点②の数値は3となる。

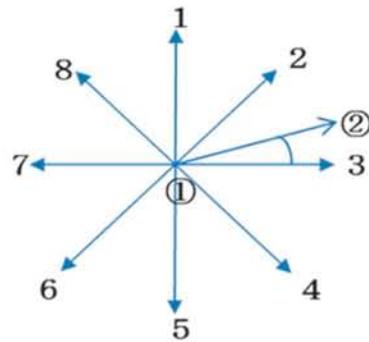


図9. 方向コードと角度

角度は式②によって求める。代表点iと次の代表点i+1の座標日から水平軸に対する角度が求めることが可能である。

$$\theta = \tan^{-1} \left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i} \right) \quad \text{②}$$

認識に使用する計算式は式③を参照。

$$D = \sum_{i=0}^{n-1} |MC_i - TC_i| \quad \text{③}$$

ただし、 $|MC_i - TC_i| > 4$ の時は

$$|MC_i - TC_i| = 8 - |MC_i - TC_i|$$

Dは総和、nは代表点の数、MCは登録済みの文字の各コード列、TCは認識したい文字の各コード列である。

3. 2 本研究の認識方式

方式3 加速度を利用した認識方式

Wiiリモコンから得られた代表点におけるx,y,z軸方向の加速度と標準文字の加速度の差をそれぞれ求める。そして、各差の絶対値の総和が最も小さくなる標準文字を見付け出すことで認識を行う方式。加速度を使うことにより文字の特徴を認識に使うことが可能である。

図10に代表点と加速度について示す。x,y,zの右に書かれている数値は加速度センサーから得られた各軸の値である。

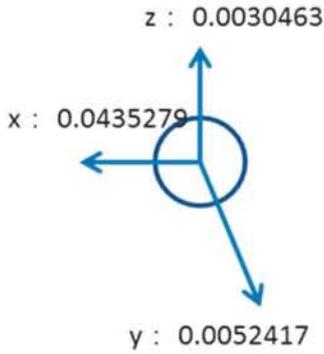


図 10. 代表点と加速度の例

認識に使用する計算式は式④を参照。

$$D = \sum_{i=0}^{n-1} |MX_i - TX| + |MY_i - TY_i| + |MZ_i - TZ_i| \quad \text{④}$$

Dは総和、nは代表点の数、MX、MY、MZは登録済みの文字の各加速度、TX、TY、TZは認識したい文字の各加速度である。

Wiiリモコンから得られた加速度の値には重力加速度が付加されている。認識に使用するに当たり、より正確な値を得るために除去する必要がある。重力加速度は一定値であるため、フーリエ変換を行うことで直流成分として検出可能。この直流成分を取り除き、逆変換を行うことで純粋な加速度を取り出すことが可能となる。

図 1 1 はWiiリモコンから得られた加速度の値をフーリエ変換した値をグラフにしたものであり、楕円で囲んだ部分が含まれていた重量加速度である[3]。縦軸がフーリエ変換した時のスペクトルを示しており、横軸が周波数を表している。x,y,z軸全てのグラフが図 1 1 のように重力加速度が検出され、左側が突出した形となっており、全ての軸のそれぞれの結果から取り除く。

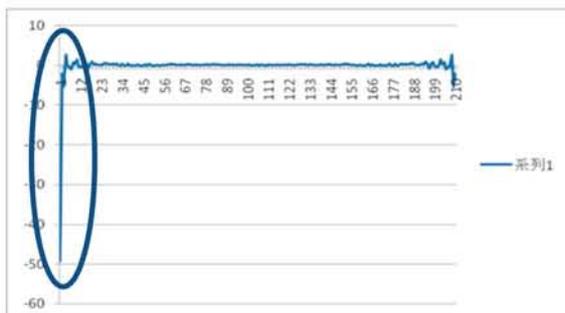


図 11. フーリエ変換をして得られたグラフ

4. 実験

4. 1 実験内容

検証はWiiリモコンで書いた文字を二種類の方法で行い、認識率を比較した。

1. 方式 1 と方式 2 の組み合わせ (以下、方式 4)
2. 方式 1 ~ 3 の組み合わせ (以下、方式 5)

認識率は認識に使用した 7 8 文字に対し、一つの文字につき十回ずつ認識を行い、一致した個数の総和から全体の数を除算して算出する。7 8 文字は数字、アルファベット、一部の記号である。辞書に登録されている標準文字は 1 文字ずつであり、以下の通りである。

0~9、a~z、A~Z、+、-、×、÷、∫、(、)、Σ、√、=、>、<、≥、≤、|、[、]

また、方式 4 と方式 5 では同じ辞書を使用しており、辞書に登録されている代表点の情報は同じである。

4. 2 ランク

ランクとは認識候補として選ばれた文字からどれぐらい近いのかを示すものである。認識が失敗していた場合、どの程度成功から離れていたかを知ることが可能である。図 1 2 の楕円で囲まれた部分がランクである。楕円の右に表示されている文字及び数字は辞書に登録されている各文字とその計算結果である。この計算結果は距離と呼び、各認識方式の計算結果を乗算している。距離が小さい文字がランクの 0 に近くなり、ランクが 0 の文字が認識結果として出力された文字となる。今回の実験ではどれだけ正しく認識されたかを集計するだけでなく、このランクの集計も行った。

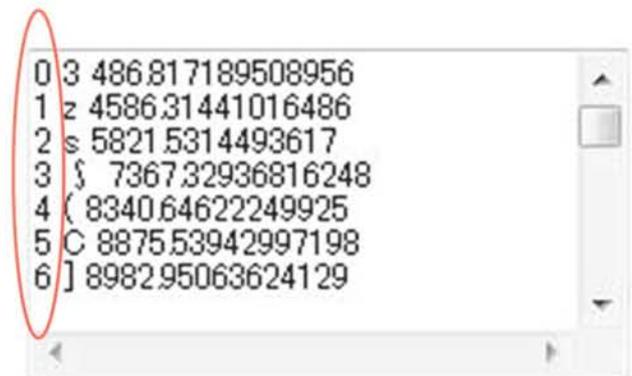


図 12. ランクの例

5. 実験結果

結果は次の通りとなった。

方式4

一致の総数：659

認識率：約85.5%

方式5

一致の総数：739

認識率：約94.7%

実験の結果、約9.2%の認識率の上昇となった。全体的にどの文字も認識率及びランクが上昇した。中でも大幅に改善された文字は0、1、j、k、O、Q、R、|と似た形が存在する文字である。

認識率の変化

0 : 5 → 9

1 : 1 → 7

j : 4 → 9

k : 3 → 8

O : 6 → 9

Q : 4 → 10

R : 6 → 10

| : 3 → 8

座標や方向コードでは似た形を判別することが難しい。例えば、jはiと判定されてしまう場合が多かった。加速度を使用した認識方式を加えることによって文字自体の特性を活かすことが出来、差別化が出来た。

対して、方式4で認識率が低かったq、[,]の3文字に関して認識率は殆ど変わらなかった。しかし、ランクは全体的に上昇していた。

ランク平均の変化

q : 約7.5 → 1.1

[: 1.3 → 1

] : 0.9 → 0.7

このことから方式5は方式4よりも認識の精度が上がっていることがわかった。

6. おわりに

本研究では、Wiiリモコンを用いた数式のオンライン認識方式について検討を行い、実験結果から有効性を確認した。また、加速度を利用した方式を加え、従来方式と比較を行い認識率が上昇したことを確認した。今後は認

識率の上昇が少なかった文字を認識出来るように改良していきたいと考えている。また、代表点の変更も行っていきたいと考えている。現在、代表点は8個で固定としているが、その決め方は1ストロークを一定間隔で区切ったものとなっている。Wiiリモコンから得られた加速度を認識に使用しているが、認識する為に必要な代表点の変更、もしくは新たな代表点の追加を行うことを考えている。図13が例である。

また、加速度を用いた方式についてより細部について検討を加えるとともに実験を行って同方式の有効性について検討を加える予定である。

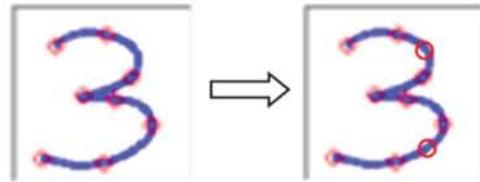


図13. 代表点を増やした例

参考文献

- [1] 酒井幸市 “画像処理とパターン認識入門” 森川出版株式会社, (2006)
- [2] 伊藤等, 中川正樹 “Hidden Markov Model に基づくオンライン手書き文字認識”, 電子情報通信学会技術研究報告, PRMU, パターン認識・メディア理解, 97(205), 95-100, 1997-07-25
- [3] 杉本真佐樹 “Wii リモコンを用いた双方向授業支援システムの開発”, 国立鳥羽商船高等専門学校, 専攻科, 生産システム工学専攻ホームページ (<http://www.info.toba-cmt.ac.jp/~ezaki/wp-content/uploads/2011/12/sugimoto-senkouka.pdf>)
- [4] 嵯峨山茂樹, 中井満, 下平博 “ストローク HMM に基づくオンライン手書き文字認識方式”, 2000 年信学技報, PRMU2000-135, pp.39-46, 2000 年 12 月
- [5] 白井暁彦, 小坂崇之, 尾崎俊介, 原央樹, 木村秀敬 “WiiRemote プログラミング” オーム社, (2009)
- [6] 松永啓佑, 村上仁己, 小池淳 “Wii リモコンを使用した記号を含む数式のオンライン認識”, 電子情報通信学会総合大会, D-9-35 (2012.3)