Vol.2010-HCI-140 No.1
Vol.2010-UBI-28 No.1
2010/10/29

iPod touch の加速度センサによる動作判別用ライブラリの構築

小瀧　陽[†1]　　笹倉　万里子[†1]

本研究は iPod touch の加速度センサの値から動作を判別するための知識を組み込んだ動作判別用ライブラリを構築することを目的とする．ライブラリを使用することでプログラミングを行うユーザは簡単に加速度センサを用いた動作判別を行うことができる．本研究で構築するライブラリでは，次の動作を判別することができる．1) 上下左右前後の 6 方向の動き，2) 本体が傾いた動き，3) 本体の傾き角度．これらの動きを判別するために，加速度センサの変化のパターンの特徴を用いた．ユーザはこれらの動きを組み合わせて独自のインタフェースを構築することが可能となる．

A library for detecting movements of an iPod touch by 3D acceleration

Akira Kotaki[†1] and Mariko Sasakura[†1]

The aim of this study is to develop a library for detecting movements of an iPod touch by 3D acceleration. A programmer can develop a system which detects movements by 3D acceleration easily. The library developed in this study can detect 1) movements toward up, down, left, right, forward or backward, 2) movements of leaning, and 3) an angle of leaning. We use particular patterns of values of 3D acceleration. A programmer can construct a new interface combine the movements.

1. はじめに

Wii リモコンや iPod touch/iPhone 等の加速度センサを内蔵したデバイスが簡単に手に入るようになり，これらを利用したシステムの開発が盛んに行われる様になってきた．加速度センサはそれが内蔵されている物体が動いた際にその物体にかかっている加速度を測定するものである．例えば iPod touch/iPhone の中でディスプレイに表示されている内容が本体の向きを変えた際に同調し変わるものは加速度センサで重力の影響を検知して行っている．

しかし，加速度から動作や距離を特定するというのは難しい．我々が以前行った iPod touch の加速度センサを使用し距離を特定しようとする研究でも精度の向上は大きな課題として残った[8]．

加速度センサで計測できるのは加速度であって速度ではない．例えば加速度センサの値からは止まっている場合と一定速度で動いている場合の区別がつかない．人は直感的にどこからどこまで動いたかという距離でものの動きを解釈すると考えられるが，加速度センサだけでは基本的には動いた距離は分からない．

他にも加速度センサによる動きの判別を困難にしている原因には以下のものがある．

- 重力の影響を受ける．
- 手のブレ等で数値が安定しない．
- 同じ操作でも人により動かし方が異なり，計測される加速度センサの値が違う．例えば右に動かした場合でも人によって微妙ながら動かす方向や距離が違う．

今後加速度センサ自体の性能が向上したとしてもこれらの問題は解決されないだろう．加速度センサの値から人が意図したと思われる動作を判別するためにはどのような動作をした時に加速度センサの値がどのように変化するかという知識が必要である．そこで加速度センサの値から動作を判別するための知識を組み込んだ動作判別用ライブラリを作成してプログラミングを行うユーザの負担を減らすことを本研究の目的とする．このようなライブラリがあれば加速度センサによって動作を判別し，それを利用したインタフェースを簡単に構築することができる．

本研究では iPhone/iPod touch(以下 iPod touch と略す) 用のライブラリを構築する．このライブラリでは以下のものが判別できる．

- iPod touch の上下左右前後の 6 方向の動き
- 本体が傾いた動き
- 本体の傾きの角度

iPod touch の特性については 2 節で述べるが iPod touch を入力デバイスとして利用することで，従来のマウスやキーボードよりも直感的な動作が可能とになると考えている．ど

†1 岡山大学大学院自然科学研究科
Okayama University Graduate School of Natural and Technology

Vol.2010-HCI-140 No.1
Vol.2010-UBI-28 No.1
2010/10/29

のようにして動作を判別するかについては 3 節で詳しく記述する．4 節でライブラリの仕様を，5 節でこのライブラリを用いたアプリケーションの例を示す．6 節で関連研究について述べ 7 節でまとめる．

2. iPod touch の特性

図 1，図 2 が本研究で使用する Apple 社製の iPod touch である．iPod touch にはマルチタッチスクリーン，3 軸加速度センサ (本稿では加速度センサと略称する)，無線通信機能が標準的に内蔵されている．これらの機能を持った携帯機器も近年多いが，その中で iPod touch には以下の特徴がある．

(1)　Apple 社により開発環境が用意されている
(2)　作成したアプリケーションは Web を通じて配布可能である

上記の様に開発，配布環境が整っているためマルチタッチスクリーン，加速度センサを利用したアプリケーションが多数存在する．特にタッチパネルを使用したアプリケーションは多く，今までキー操作によって行われてきた操作をより簡単なものにしている[1),2)]．しかし加速度センサを用いたアプリケーションには本体の向きを判別することや，傾きによる操作はあるものの，より複雑な動き等を実現しているものはあまりない．そこで動作ライブラリを構築すればより加速度センサを利用したアプリケーションが簡単に作ることができるようになると期待できる．

3. 動作の判別

本研究で構築するライブラリでは加速度センサの値の変化から iPod touch がどのように動いたかの判別を行う．図 1，図 2 に示す様に iPod touch を縦長に持った時に水平方向が x 軸，垂直方向が y 軸，前後方向が z 軸である．

図 3 が加速度センサから取得した数値の変化の例である．これは 0.1 秒ごとに加速度を取得したもので 20 秒間の加速度の変化の様子である．実際の加速度の数値はとても小さいため視覚的にみやすいように定数倍している．また 3.2 節で詳しく述べる基準値との差をとっているため基準となる黒いラインに近い場合が手に持ち静止している状態であり，数値が大きく変化しているのが iPod touch を動かした部分である．

3.1 判別動作の種類

検出した加速度から動作を判別する際に加速度の数値の変化からどのように動いたか判別する．動作には判別しやすい動作と判別しにくい動作がある．以下に判別しやすい動作を

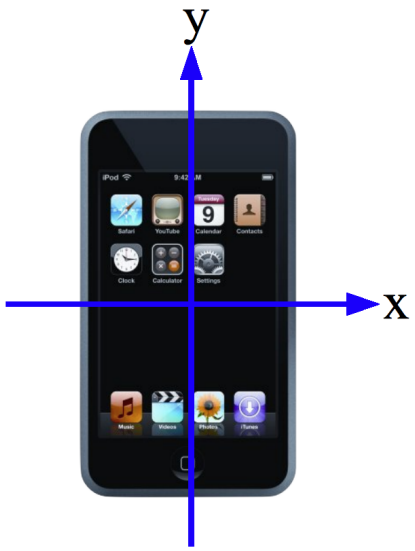

図 1　iPod touch に対する x,y 軸の割当

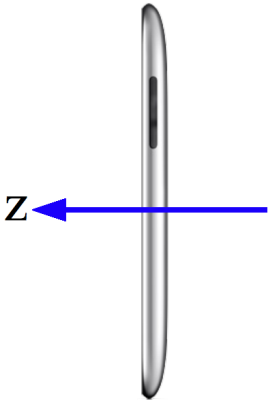

図 2　iPod touch に対する z 軸の割当

挙げる．

(1)　重力を受ける軸が変化する動作
本体の向きが変わった場合重力を受ける軸が変わる．そのため本体の向きを変える動作はとても分かりやすく判別できる．

(2)　誰が動かしても同じ波形を得ることが出来る動作
抽象的な動作は人により解釈が異なるため判別を行うのがとても難しい．逆に誰が動かした場合でも同じような加速度の変化を得ることが出来る動作は判別が容易である．例えば右方向に動かす動作がこれにあたる．

(3)　加速度に特徴がある動作
加速度に特徴があり他の動作とはっきり区別がつく動作というのはとても判別を行いやすい．その変化があった場合には即その動作と結びつけることができるからである．例えば机の上で iPod touch を裏返す動作がこれにあたる．

判別しやすい動作と判別しにくい動作は反対の関係にあるものが多い．以下に判別しにくい動作を挙げる．

Vol.2010-HCI-140 No.1
Vol.2010-UBI-28 No.1
2010/10/29

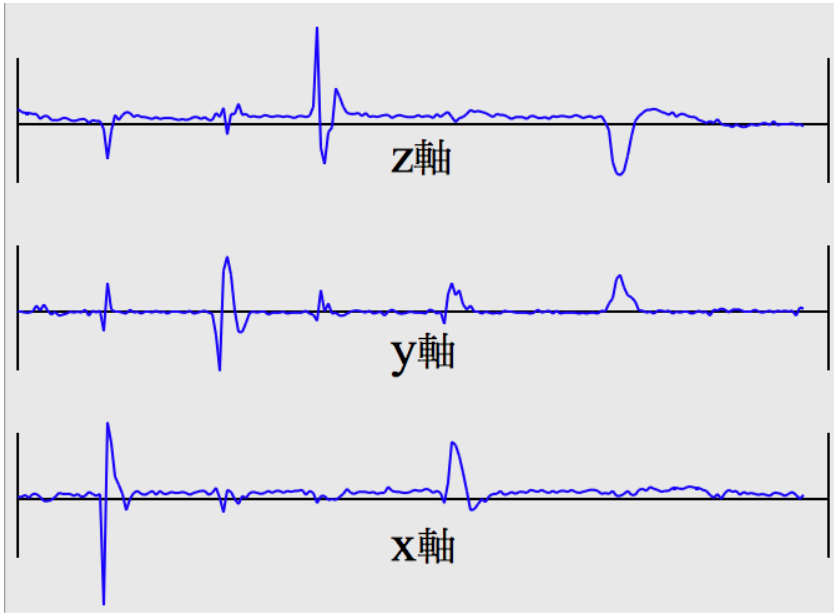

図 3 加速度の値の例 (20 秒間)

(1) ゆっくり動かす動作
ある程度の動作速度がない場合は加速度に大きな変化が見られない．これは 6 方向にゆっくり動かした場合も判別できないということである．この問題はユーザの使用方法に影響されるため解決するのがとても難し．そのため使用するユーザにあらかじめある程度の速度で動かすように協力して頂く必要がある．

(2) 素早く複雑に動かしすぎる動作
(1) の問題とは逆に素早くいろいろな方向に動かした場合加速度の値が変化し続けるため一つ一つの動作を区別するのが難しい．操作する際には一つの動作をするごとにある程度の間が必要である．しかしこの問題もユーザによってどこまでが可能でどこまでが不可能なのかを判別するのは難しい．そのためインタフェースに慣れる時間がある程度必要になると考えられる．

(3) 他の動作との判別が難しい動作
例えば手にもって振る動作と目の前で円を描くように動かす動作は加速度センサの値が非常に似かよったものとなるため判別できない．

(4) 人により動かし方が違う動作
例えば「投げる」動作は人によりどのように投げるかが異なるため判別できない．

これを踏まえ現時点では判別する動作を以下の 3 種類に決めた．これらはユーザが iPod touch を動かすスピードという問題点がある以外は誰が動かした場合でも共通に判別できると考えている．

- 各軸に対する角度
- 上下左右前後の 6 方向の動作
- 各方向に傾けた場合の動作

各軸に対する角度は加速度の値を数値変換で求めるため簡単に求めることができる．しかし上下左右前後に動かす動作と傾けた動作は加速度の変化のパターンを見る必要がある．そのパターンについては 3.2 節で詳しく説明する．

他の動作にはこれらの要素を複合的に組み合わせることで判別を行う．ライブラリとしては上記の 3 種類の動作を判別するだけなので，組み合わせでの動作の判別はライブラリを使用しているアプリケーションに一任する．そのため新たな動作の組み合わせはアプリケーションごとに自由に決めることができる．

3.2 判別方法

この節では上下左右前後に動かす動作と傾けた動作の判別のための加速度の変化のパターンについて説明する．例えば左に動かすというのは iPod touch を手に持ち左に図 4 のように水平移動することである．また左に傾けるというのは図 5 のように手首を軸にして傾けることである．

図 6 の赤いラインと加速度が 0 である黒いラインの差が受けている加速度の大きさである．このようにずれるのは重力の影響を受けるからである．iPod touch は手に持ち操作するため当然厳密に水平に保つことは不可能である．そのため全ての軸に対して重力の影響を考慮するために基準点というものを考える．持つ角度で各軸にかかる重力は変化するため基準点の設定には一定時間加速度に大きな変化がない場合その時の値を基準値とする．図 7 の黒い横のラインが 0 であり，青いラインが図 6 の加速度の数値から基準値を考慮した結果である．加速度の変化がない場合には加速度の数値と基準からの差が 0 に近いことが分かる．このようにして本体の角度が変化する度に基準点を設定する．

上下左右前後の動作と傾きの判別するために以下のポイントに注目する．

(1) 静止状態の判別

Vol.2010-HCI-140 No.1
Vol.2010-UBI-28 No.1
2010/10/29

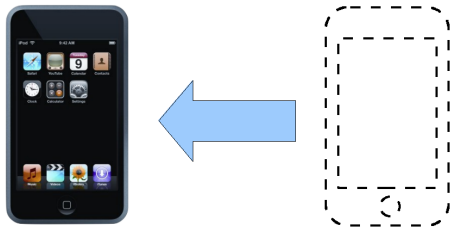

図 4 左に動かす例

図 5 左に傾ける例

図 6 基準補正なし

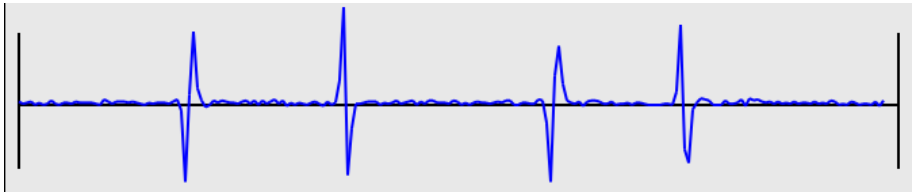

図 7 基準補正あり

上下左右前後で判別するため，iPod touch をどのように持っているかで x,y,z 軸の対応する方向が変化する．そのため現在ユーザがどのように持っているかを判断する必要がある．iPod touch は重力の影響を受けるため，x,y,z 軸の中で重力の影響を受けている軸を基準として考えることで判断する．

(2) 静止状態からの変化方向

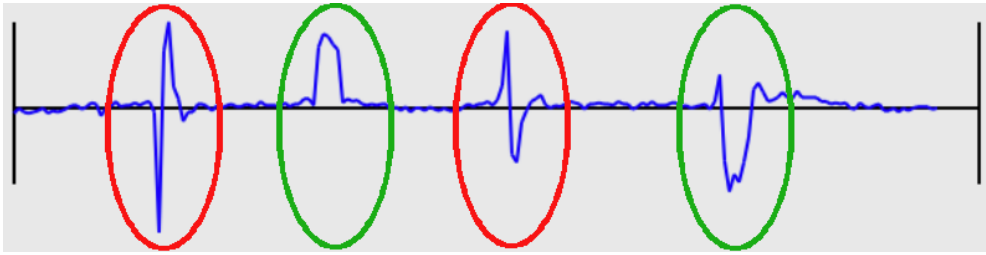

図 8 動かした場合の変化

iPod touch を動かした場合の数値の変化に注目する．静止状態から動かした場合各軸の加速度の変化と本体の向きからどちらに動いたかを判別する．

(3) 動作開始後の変化方向

動作には上下左右前後の動きと傾きを区別する必要がある．図 8 が上下左右前後の動きと傾きの数値変化の違いである．赤い楕円で囲まれた部分が上下左右前後に動いた場合の変化であり，緑色の楕円で囲まれた部分が本体を傾けた場合である．図 8 で分かる様に 6 方向の動きでは加速度が最初に変化した方向と反対側の方向にも加速度を検知しているのが分かる．しかし傾けた場合には加速度の変化があった後基準点までしかもどっていない．この違いを判別の基準として使用する．

(4) しきい値の設定

加速度センサは手のブレでも少量の加速度を検知してしまうため，純粋な変化だけに注目してしまうと常に何かしらの動作をしていることになってしまう．そのため微妙な変化を無視するためにしきい値を設定することでその値を超えた場合に変化したと判別する．

4. ライブラリとしての機能

今回判別するプログラムをライブラリとしてまとめることで今後このインタフェースを使用する場合に使用者は動きの判別はライブラリに任せて結果だけを使用することが可能である．ライブラリの機能を以下にまとめる．

(1) setInit

測定間隔を指定する．この時間間隔で以下の処理結果を返す．

(2) getAccele

加速度センサから取得した値を setInit で指定した時間間隔ごとに 3 軸それぞれの値を引数

Vol.2010-HCI-140 No.1
Vol.2010-UBI-28 No.1
2010/10/29

表 1 メソッド名 ： (返り値) メソッド名:(型) 引数

メソッド名	詳細
(void)setInit:(float)time	時間間隔を指定する
(float)getAccele:(int)count	時間感覚ごとに加速度センサの数値を返す
(NSString)getMove:(int)count	動いた方向を返す
(int)getAngle:(int)count	角度を返す
(NSString)getLean:(int)count	傾いた方向を返す

に指定された回数だけ返す．数値そのものを取得するものを用意したのはライブラリとして用意しているもの以外に実現できる動作がある場合にアプリケーション作成者が加速度を使用できるようにするためである．

(3) getMove

上下左右前後の 6 方向の向きに動いた方向を引数に指定された回数だけ返す．判別した結果を up, down, left, right, forward, back で区別する．この方向は使用者がどのように iPod touch を持っていたとしても使用者から見てどちらの方向かを示すものである．

(4) getAngle

各軸に対する角度を引数に指定された回数だけ返す．水平においた状態を 0 °とし 180 °までを返す．

(5) getLean

各軸ごとに傾いたかを判別し引数に指定された回数だけ方向を返す．判別した結果を up, down, left, right, forward, back で区別する．方向の基準は getMove の場合と同じである．

表 1 が機能をメソッドにしたものである．言語は Objective-C である．開発者は加速度センサに関する設定を行う必要がなくなり，動作を知りたい場合にこれらのメソッドを呼び出すだけでよい．

5. 実 行 例

本節ではライブラリを使用したアプリケーションの例を示す．

(1) 傾きによる画像操作

図 9，図 10 は我々が作成した加速度の変化を操作に用いたアプリケーションの例である．表示されている画像の位置の変更を iPod touch を横に傾けることで行う．また画像の拡大，縮小の操作を前後に傾けることで行う．オープンキャンパス等で実際に一般の方に操作して頂いたが最初はどれくらい動かせばいいか分からないものの次第に慣れて思う様に動かすことができていた．ボタン等の操作を必要としないため分かりやすく感覚で操作することができる．

図 9 静止時の画面

図 10 傾けた場合の表示位置の移動

(2) 判別結果の組み合わせによる認証

判別できる上下左右前後の組み合わせで認証を行うものである．パスワードの様に予め人ごとに認証する動作を登録しておく．それにより認証を行う際に iPod touch を上下左右前後の一方向に順番に動かすことで認証する．図 11 では右→上→右→下の順番動かすと登録している場合の例である．認証する際には図 11 の青い矢印の順番で動かすと成功する．しかし赤い矢印の様に決められている動作以外の方向に動いた場合は認証に失敗する．

6. 関 連 研 究

従来の動きを検出するインタフェースとしては，多くのセンサを使用し判別するものや，カメラを使用し画像解析を行うもの等がある[3),6)]．しかしこれらは特別な機材を必要とし，一般的に容易に使用できるものではない．しかし加速度センサを内蔵した iPod touch なら誰でも入手可能であり，持ち運びも簡単であるため誰でも気軽に使用することができる．携帯機器なので手に持ち手の動きだけしか判別することは出来ないがインタフェースとして使用するには十分だと考えられる．同様の用途で iPod touch 以外の加速度センサを利用した様々なインタフェースも研究されている[10)]．

また近年では Wii コントローラーを用いたインタフェースも研究されている[4),5),7)]．これらも手に持ち動き等を感知することができる．

iPod touch にはディスプレイがあるため入力デバイスとしてだけではなく出力デバイスとしても用いる事が出来る．例えば[8)] では，iPod touch を動かす事で iPod touch に表示

Vol.2010-HCI-140 No.1
Vol.2010-UBI-28 No.1
2010/10/29

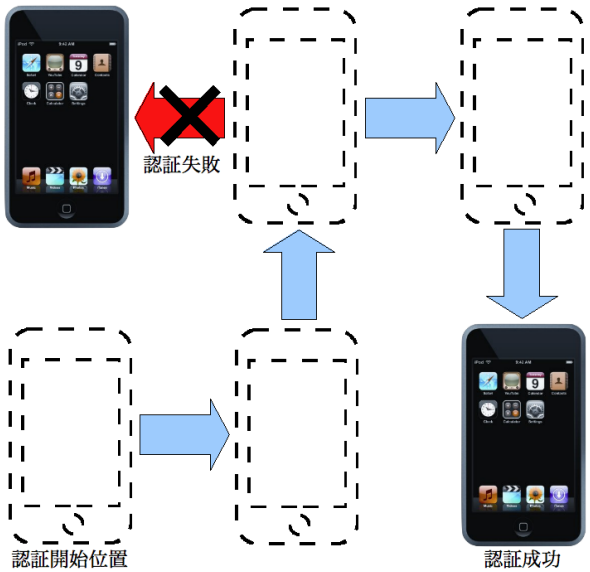

図 11 動作による認証の例

する画像を動かす事を試みている．

7. おわりに

本稿では加速度を使用した動作を判別するインタフェースとそのライブラリについて報告した．報告したライブラリでは以下のことができる．

- iPod touch 本体に対して上下左右前後の 6 方向の動きの判別
- 本体が傾いた動きの判別
- 本体の傾きの角度の判別

このライブラリを利用すれば，ユーザは現在の iPod touch の向きや使用者ごとによる動きの違いを気にせず iPod touch の動作を入力とするプログラミングを行うことができる．

今後は現在のライブラリとしてのシステムの向上の他にもこのライブラリを用いたアプリケーションの開発を行っていきたいと考えている．

現在幾つかの改善すべき点がある．以下に改善点と解決案を挙げる．

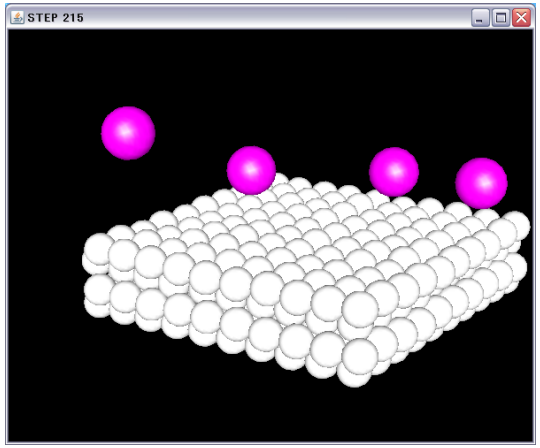

図 12 3D シミュレーション

(1) 判別動作の少なさ

現在は確実に動作を判別できるようにという考えから判別する動作は限られている．しかし汎用性やインタフェースとしての使いやすさを向上させるためには多くの動作を判別できることが求められる．6 方向の動きから複合的に判別できる動作以外にも特徴的な動作を判別したいと考えている．新しい動作には誰が動かしても同じような加速度を取得できることと判別結果が正確であることが求められる．これらを満たすためにはアルゴリズム自体の見直し等が考えられる．例えば，本研究で開発したライブラリでは「投げる」動作を認識できないが「投げる」ことでメールを送るインタフェースの研究もすでに行われている[9]．

(2) インタフェース使用の際の制限

iPod touch をゆっくり動かしすぎたり，早く動かしすぎたりした場合の動作判別の失敗はプログラム内での解決は難しいと考えている．この問題はインタフェース使用者に依存しているため，初めてこのインタフェースを使用する人が最初から正確に使用することは難しいだろう．ある程度使えば感覚的に理解することが可能にはなることを考えるとゲーム感覚でどの程度の力で動かせばいいか理解できるようなアプリケーション等があってもいいかもしれない．

このライブラリにさらに機能を追加し同じ研究室内にて研究している図 12 のような 3D モデルの操作に使用することを検討している．3D モデルの操作ではどのように操作すれば

Vol.2010-HCI-140 No.1
Vol.2010-UBI-28 No.1
2010/10/29

よいかが分かりにくいため，iPod touch を実際に回すことで回転を行うことや，断面表示のための断面の指定を iPod touch の角度で指定することを考えている．また iPod touch のディスプレイに断面図を表示する等 iPod touch とコンピュータのディスプレイを合わせて見れる様な使用方法も検討中である．

また他にも本研究のインタフェースとして機能を活かせる新たなアプリケーションを考えていきたいと考えている．

参 考 文 献

1) Barrass, S., Schaffert, N. and Barrass, T.: Probing preferences between six designs of interactive sonifications for recreational sports, health and fitness, Proceedings of ISon 2010, 3rd Interactive Sonification Workshop, pp.23-29 (2010).
2) Kim, J.-S., GraV canin, D., Matković, K. and Quek F.iPhone/iPod touch as input devices for navigation inimmersive virtual environments, IEEE Virtual Reality 2009, pp.261-262 (2009).
3) Mistry, P., Maes, P. and Chang, L.: WUW - ware Ur world: a wearable gestural interface, in CHI EA '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems, pp.4111-4116 (2009).
4) Santos B. S., Prada, B., Ribeiro, H., Dias, P., Silva, S. and Ferreira, C.: Wiimote as an input device in Goole Earth visualization and navigation: a user study comparing two alternatives, 14th International Conference Information Visualisation, pp.473-478 (2010).
5) Sheridan, J. G., Price, S. and Pontual-Falcao, T.: Wii remotes as tangible exertion interfaces for exploring action-representation relationships, Whole Body Interaction 2009, A SIGCHI 2009 Workshop (2009).
6) Tamaki, E., Miyaki, T. and Rekimoto, J.: Brainy hand: an ear-worn hand gesture interaction device, in CHI EA '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems, pp.4255-4260 (2009).
7) Wingrave, C. A., Williamson, N., Varcholik, P. D., Rose, J., Miller, A., Charbonneau, E., Bott, J. and LaVIola Jr., J. J.: The Wiimote and beyond: spatially convenient devices for 3D user interfaces, IEEE Computer Graphics and Applications, vol.30, no.2, pp.71-85 (2010).
8) 小瀧陽，笹倉万里子：携帯機器を利用した仮想虫眼鏡の提案，第 23 回人工知能学会全国大会 (CD-ROM), 高松, 2009.
9) 忍頂寺毅，長谷川慎，落合桂一: 投げメール: 位置情報と身体性を利用したコミュニケーション，WISS 2009 (2009).
10) 松尾 賢治，奥村 文教，橋本 眞幸，小池 淳，久保田 彰，羽鳥 好律：加速度センサを用いた携帯電話向けの腕振り行動認証に関する検討， 電子情報通信学会総合大会講演論文集, pp.541(2006).