

ウェアラブルコンピューティングのための 手足を使った状況依存コマンド入力手法

山本 哲也^{†1} 庄司 武^{†2} 寺田 努^{†3}
塚本 昌彦^{†3} 義久 智樹^{†4}

ウェアラブルコンピューティングが一般に広がるにつれ、ユーザが自分のもつ情報と環境情報を利用したサービスを常に受けることにより、ユーザは様々な状況で多くの情報を扱うこととなる。また、従来のデスクトップコンピューティングと異なり、ウェアラブルコンピューティング環境でサービスを受ける際には、靴をもつていて右手が使えないなど様々な状況を想定する必要があるため、状況の変化を考慮したインタフェースデザインが必要である。本研究では、特に状況の変化を考慮した入力方式に着目し、両手両足のジェスチャ入力メニュー等に入力を行うような環境において効率的な入力手法を自動的に選択するインタフェースを提案する。本研究ではまず実験により、両手両足の加速度センサによる入力から状況に応じてどのような特性をもつのかを明らかにし、状況の変化を考慮した入力方法を実現する。

A Context Aware Command Input Method using Hands and Feet for Wearable Computing

TETSUYA YAMAMOTO,^{†1} TAKESHI SHOJI,^{†2} TSUTOMU TERADA,^{†3}
MASAHIKO TSUKAMOTO^{†3} and TOMOKI YOSHIHISA^{†4}

As wearable computing become popular in the world near future, people must handle abundant information in many situations because they will get variety of services which use their profiles and environmental information all the time. In order to manage such large volume of information, we need more convenient methods for command input to computers suitable in various situations than conventional input devices such as handy mouse with stick. Thus, we designed situation dependent commands for wearable computing. We use bracelet-like devices and shoe-like devices which embed accelerometers to get user's gesture input. In this paper, we show characteristics of gesture input for hands and foot operations in various situations, and then propose several situation dependent methods for command input.

1. はじめに

コンピュータの小型化、高性能化に伴い、携帯電話やノート PC による情報通信は日常的な行動となっている。将来、ウェアラブルコンピューティング、ユビキタスコンピューティングが一般になれば、ユーザは自分のもつ情報と環境の情報を利用したサービスを常

に受けることによる。そのような環境では従来のデスクトップコンピューティング環境と異なり、ユーザは様々な状況で多くの情報を扱うこととなる。例えば図 1 に示すように、街角を歩いている際にお店からの情報を受けて選択画面が HMD に表示され、詳細を表示するか否かなどの選択をおこなう場面が増えてくると予想される。このように常に情報が表示されるようなデバイスを利用する場合、様々な状況でも簡単に入力が行える必要がある。状況の例としては、歩いているときの他に、カバンを持ったり、筆記用具を持っており片手が使えないといった状況も想定されるため、体の中で利用できる部分が限られている際でも効率的な入力を行える方法が望まれている。そこで本研究ではそのような制約のあるときでも、状況の変化を考慮した最適なインタフェースが常時使えるような入力

†1 神戸大学大学院自然科学研究科
Graduate School of Science and Technology, Kobe University
†2 オムロン株式会社
OMRON Corporation
†3 神戸大学大学院工学研究科
Graduate School of Engineering, Kobe University
†4 大阪大学サイバーメディアセンター
Cybermedia Center, Osaka University



図 1 表示画面のイメージ

行うことを目的とする。

ウェアラブルコンピューティング環境では、ユーザはコンピュータを常時装着することで、これまではコンピュータを使用するとして想定されていなかった場所や状況でもコンピュータを使用できるようになり、その使用環境も変化する。このような環境では、システムに対して何かを入力したいときにすぐ利用できる入力インタフェースが望ましく、入力デバイスも常時装着して使用することが有効であると考えられる。装着型のデバイスを利用した入力としてはジェスチャ入力が一一般的である。ジェスチャ入力は体の様々な部分の動きを使って行えるが、動きの機敏さや可動範囲から手足が最も利用しやすい。また、日常的に使うためには、装着による生活への影響が軽微である必要がある。したがって、本研究は従来から身につけている靴や、アクセサリとして使用されている腕輪が日常生活への影響が少ないと考え、靴型と腕輪型のインタフェースを使用する。両手両足にこのようなデバイスを身につけてジェスチャ入力を行うようなインタフェースを想定し、ユーザ状況に応じて優先的に使用するデバイスを変更するような仕組みを提案する。

以降、2章ではセンサを用いた入力インタフェースについての関連研究について述べる。3章では状況の変化と動作の制約について述べ、4章ではシステム的设计と実装について記す。5章ではジェスチャ入力の認識精度と時間についての評価実験とその結果を整理し、6章で状況の変化を考慮したコマンド入力手法について検討する。7章でまとめと今後の課題をについて述べる。

2. 関連研究

装着型デバイスを利用したジェスチャ入力インタフェースについて、さまざまな研究がなされている。装着時に手や指を覆うことがなく、また装着による煩わしさもほとんどないデバイスとして、腕輪型デバイスと靴型デバイスがある。

GestureWrist³⁾ は加速度センサとバンドの内側に、

静電容量を測定するための電極を組み込んだ腕輪型のデバイスで、腕の向きと手の形状を検出できる。しかし、GestureWrist で使用している静電容量は、電極と接触する腕の部分で測定しており、腕を動かしたりものをつかむなどの動作をすることによって装着位置がずれると、同じ手の形状でも測定できる値が変化してしまうという問題がある。

Dancing Shoes⁴⁾ では、靴に圧力センサなど様々なセンサを取り付けた靴型デバイスで、動きを解析することで動きに合わせインタラクティブに音楽を自動生成できる。しかし、日常生活での足動きには制限があり、跳ぶジェスチャなどを日頃から行うのは無理であり、より日常的に使える足のジェスチャが必要である。

GesturePendant⁵⁾ は、ペンダント型のデバイスの赤外線カメラを組み込み、手の動きをカメラで撮影し、画像処理を行うことでジェスチャを認識する。しかし、カメラで撮影できる範囲でジェスチャを行わないと入力できないため、荷物を持っているときなど、手が自由にならない状況では、入力が行えないという問題がある。

これまでの研究は、デバイスの装着性について考慮したものが多いが、ユーザがジェスチャの入力を行う際の状況の変化を考慮しているものは少ない。一方、日常生活で常時入力インタフェースを利用するにあたっては、状況によって腕や手の動作が制限されてしまい、ユーザが望む操作を行えなくなる状況は頻繁に生じると考えられる。庄司らによる使用環境に考慮した入力インターフェース¹⁾ においても、それぞれの状況による変化を考慮に入れた操作モードの変更方法までは提案されておらず、足での入力についても検討がなされていない。本研究では、デバイスの装着性だけでなくユーザの状況変化も考慮に入れ、動作の制限に応じたジェスチャと操作の割り当てを行う点で関連研究とは異なる。

3. 状況の変化と動作の制約

ウェアラブルコンピューティングでは、日常生活の中でコンピュータを利用することから、ユーザの状況が様々なに変化する。例えば、朝起床してから夜就寝するまでを考えてみる。朝食をとった後、自転車で駅まで行き、電車に乗って街に買い物に行き、街中を散策しながらお店にいたりカフェやレストランに行く。家に帰り夕飯をつくり、本を読み風呂に入って床につく。このように、ユーザの状況は移動や行動によって変わっていく。その変化により、ユーザの手や足の可動に関する制約も変化するため、手足がどのように制

約を受けるか考える必要がある。いくつかの状況とそ
の際の手足の制約に関して下記に示す。

椅子に座っている状態： 座っている場合は、両手両
足ともに自由に動かすことができる。座っている
状態は他からの影響を受けにくいので、細かな動
作の認識が可能である。しかし、座っている状態
でも、煙草を吸っていたり筆記具を持っていると
きや、本を読んでいるときなど、比較的長い間、
片手もしくは両手が使えない状況がある。状況に
よっては、ものを持っていてもジェスチャができ
る場合とできない場合があり、ものを持っている
からといって必ずしもジェスチャによる入力が不
可能ではない。

立っている状態： 立っている場合も、基本的に座っ
ている時と同様に両手両足を自由に動かすことが
できる。ただし、足のジェスチャの場合、どちら
かの足に体重を残す必要があるため、片足ずつし
か用いることができない。また、鞆や荷物をもっ
ている場合は、片手もしくは両手が使えない状況
になる信号待ちや電車が来るのを待っているとき
この状態にあたる。

歩行中： 何も持っていないときには、両手は自由に
動かすことができるが、歩いているときに生じる
振動が認識に影響を与えると予想される。足は常
に動いているために細かいジェスチャ入力はでき
ないが、通常の歩いている動きとは異なる動きを
することでジェスチャ入力が可能である。

電車の中： 上記の椅子に座っている状態や立って
いる状態と基本的には同じであるが、電車の振動や
出発、停止の影響がユーザへ伝わるために、細か
な動作では認識に影響があると予想される。また、
つり革を持っているときには片手が制限され、周
囲の人に迷惑とならないようなジェスチャで入力
する必要があるなどが異なる要素である。

自転車に乗っているとき： 自転車に乗っている時は、
両手はハンドルを持っており、両足は自転車をこ
いでいる。ジェスチャによっては手で入力するこ
ともできると考えられるが危険であり、フィード
バックが視覚であればさらに危険であるため、入
力すること自体が好ましくない。

食事中： 食事中は椅子に座っており両手両足を使う
ことができるが、マナーの点からあまり手のジェ
スチャを使うのは好ましくない。足はテーブルに
隠れており、使うことができると考えられるが、
食事に集中した方がよいと思われる。

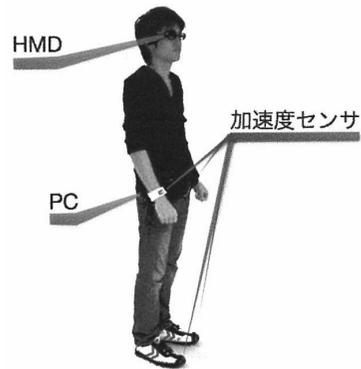


図 2 装着図

3.1 動作の制約

上記で述べたように、状況により手足のジェスチャ
入力に制約が発生する。立っているときから歩きだす
ような状況の変化だけではなく、ものを持ったり離し
たりすることで、入力できるジェスチャが変化する。
そのため、どの手が使えるかを自動的に手や足につけ
たセンサだけで判断することは難しく、どちらの手が
使えるかを伝える必要がある。

また、ほとんどの場合において、手に制約があるこ
とが多く、歩いているとき以外には足の制約は少ない。
ただし、足による入力が手ほどの繊細さを担保できる
かどうかは不透明であり、手が使えるときには手をつ
かつた方がよいと予想される。

さらに、手と足どちらのジェスチャ入力に関しても、
周囲に影響を与えないよう小さな動きであることが望ま
しい。

4. システムの提案と設計

ジェスチャを認識するために装着するデバイスとし
て、装着性を考慮し腕輪型と靴型のデバイスを用いる。
ジェスチャの認識は、デバイスに装着した3軸加速度
センサで行う。この加速度センサを搭載した腕輪型デ
バイスと靴型デバイスを両手両足に装着することで、
両手両足を使ったジェスチャの入力を可能とする。
図2にシステムを装着した様子を示す。

4.1 想定するタスク

HMDを使って常に画面が目の前に表示されている
場合を想定しており、従来のデスクトップアプリケー
ションを操作するというよりは環境からの情報閲覧や
メール閲覧などの受動的なアプリケーションを想定し
ている。そのため、複雑なポインティング等を行うの

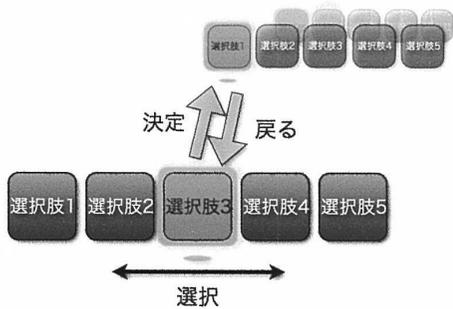


図3 メニューのイメージ

ではなく、二択もしくは複数の選択肢の中から選ぶような入力か、複雑な操作は階層的に並べたメニューで行えるようなものを想定する。そこで、入力するコマンドは、選択肢を選ぶコマンドと決定と戻るコマンド程度があれば十分となる(図3)。

4.2 ジェスチャ

3章で述べたように、ジェスチャはなるべく小さな動きであることが望ましい。また、前節で述べた選択肢を選ぶコマンドは選択肢が多いときには、連続的な値を利用する方が自然である。静止時は加速度センサにかかるのは重力加速だけであることを利用することにより、センサの角度による入力が可能である。

4.2.1 手の入力

手の動きの中で、見た目の動きが小さく加速度の変化が大きいのは、手首を中心に腕をひねるジェスチャである。腕のひねりは、図4のように手首をやや水平に近づけることで、重力加速度を利用したセンサの角度による入力が可能である。また、素早くひねることで、自分の体に向けてひねる内向き、体と反対に向けてひねる外向きの2種類のジェスチャ入力が可能である。この動きは静止時でも歩いているときでも使うことができる。なお、どちらの状態の時でも使うことができるため、歩いているときや止まっているときの選択方法として入力することもできる。

選択：手首を水平にしてひねる。

決定：手首を内向き、もしくは外向きに素早くひねる。

4.2.2 足の入力

足の動きでは、つま先を横に振るジェスチャが動作が小さく入力できる。また、つま先を上げたり、かかとを上げたりすることにより重力加速度を利用したセンサの角度による入力が可能である。(図5)しかし、座っているときや、立っているときは利用できるが歩いているときにはセンサの傾きを利用することは困難である。歩いているときには自動で選択肢が一定のテ

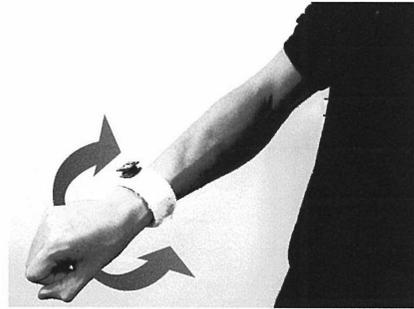


図4 手のジェスチャ入力



図5 足のジェスチャ入力(静止時)

ンポで動くなどの工夫が必要である。歩いているときでも簡単なジェスチャ入力であれば可能であり、目立たない動作として、前に足を出すときに軽く前に蹴る方法を採用した。

選択(静止時)：つま先を上げる、つま先を下げる。

決定(静止時)：つま先を横に振る。

決定(歩行時)：軽く前に蹴る。

4.3 システムの実装

腕輪型デバイスと靴型デバイスの両方に当研究室で開発した無線加速度線センサデバイスのMoCoMi-Chip⁶⁾を使用した。腕輪型デバイスでは、手の甲の側にMoCoMi-Chipを取り付け、靴型デバイスでは靴のつま先の上に取付けた。両手両足の4つの加速度センサのデータを50Hzでサンプリングし、PCで受信し認識を行う。HMDにはscalarのTeleglass T3もしくは島津製作所のDATA GLASS 3/Aを用いた。PCにはSONY社のvaio type-Uを用いた。システムのハードウェア外観を図6に示す。

ジェスチャの認識には、手と足の入力のタイプや状況によって、それぞれ以下のような手法を用いた。



図 6 PC と MoCoMi-Chip

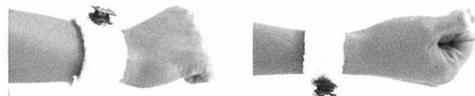


図 7 最大と最小の重力加速度

手による選択

手首を水平にしてひねることで重力加速度を利用したセンサの角度による入力を行う。手の甲と鉛直方向の加速度を使い、図 7 の左と右の状態が重力加速度が最大のとときと最小のとときであるので、これら値の間を任意の数で区切ることによって離散的な値の入力を行う。

手による決定

手首を内向き、もしくは外向きに素早くひねることで決定とする。腕を素早くひねることで、手の甲を動かす方向の加速度が動き始めと終わりの加速度の差がある閾値以上になったかどうかで判断する。また、そのときのピークのの出入りで内向きか、外向きかを判断する。

足による選択（静止時）

つま先の方向で重力加速度を利用したセンサの角度による入力をおこなう。足の前後方向の加速度をつかい、足を地面に置いているときに中心の値となり、つま先を上げたとき、かかとを上げたときに重力加速度が最大と最小となる。手の場合と同様にこの値を任意の数で区切ることによって離散的な値の入力を行う。

足による決定（静止時）

つま先を左か右に振ることで決定とする。手による決定と同様に、つま先を素早く横に振ることで、足を動かす方向の加速度が動き始めと終わりの加速度の差がある閾値以上になったときになったかどうかで判断する。また、そのときのピークのの出入りで右向きか左向きかを判断する。

表 1 座っている時の結果

選択・決定	正解率 (%)	時間 (s)
右手・右手	91.6	70
右手・左手	94	59.1
右手・右足	94.7	62.3
右手・左足	93.7	65.8
左手・右手	95.7	56.7
左手・左手	92	68.1
左手・右足	97	62.7
左手・左足	94	60.8
右足・右手	96	87.4
左足・右手	96	85.5

足による決定（歩行時）

歩行中ジェスチャ認識には DP マッチングを用いた、DP マッチングでは入力された加速度データとあらかじめ用意しておいた基準データ同士の類似度を比較することでジェスチャを認識する。認識したいジェスチャのデータと静止時のデータを比較して認識したいジェスチャのデータとの類似度が高ければ決定とする。歩行中は 4.2.2 で述べたように、軽く足を前に蹴る動作を認識させる。

5. ジェスチャ入力の認識精度と時間

5.1 座っている時

メニューへの入力として、両手両足のどのジェスチャを使うか決定するために、それぞれのジェスチャの認識精度と入力時間を調べた。選択のジェスチャでは 5 段階に分けており、腕の角度で 1～5 までの数字を選択する。それをランダムにでてくる数字と同じ数字に合わせて、決定のジェスチャで決定する。ただし、決定のジェスチャである、手のひねりの内側外側と足の振りの左右はどちらでもよい。これを座っている時に 50 回連続で行ったときの正解率と時間を評価する。被験者は 20 代前半の男性 5 人で、全員利き手は右手であった。よく練習してもらった後に実験を行った。表 1 に結果を示す。

どの正解率も 9 割を超えており、大きな差はないと考えられるが、選択と決定が同じ手である場合は正解率、時間共にやや正解率が低く時間がかかっている。右手と右足を決定に使っているときには、左手や左足が決定のときに比べてやや正解率が高い。これは利き手が決定のときに比べてやや正解率が高い。これは利き手が影響しているものと考えられる。時間は決定に足を使うときに比べて、右手・左手又は左手・右手の組み合わせが速く入力できている。選択に足を使うと手を使うときに比べて時間がかかっている。よって、以下のような特徴がある。

- 選択と決定は別のデバイスを使う方が良い。

表 2 歩いている時の結果

選択・決定	正解率 (%)	時間 (s)
右手・右手	87	83.3
右手・左手	87.6	64
右手・右足	85.6	84.8
右手・左足	86.6	84.5

- 選択と決定は手を使う方が良い。
- 決定は利き手の方が良い。

5.2 歩いている時

歩いているときのジェスチャ入力について、それぞれのジェスチャの認識精度と時間を調べた。選択のジェスチャでは5段階に分けており、腕の角度で1～5までの数字を選択する。それをランダムにでてくる数字と同じ数字に合わせて、決定のジェスチャで決定する。これを50回連続で行ったときの正解率と時間を評価する。被験者は20代前半の男性4人で、全員利き手は右手であった。よく練習してもらった後に実験を行った。表2に結果を示す。

正解率はどれもあまりかわらず、座っているときにくらべてやや低い値となっている。時間も全体的に長くかかっており、左右の手を使う時が1番速い。

6. 状況の変化を考慮したコマンド入力手法

足よりも手を使う方が良いことがわかったが、状況によってはかならずしも両手を使えるとはかぎらない。また、前章より、手よりも足の方が目立たずに入力することができ様々な状況に対応しやすい。そこで、それぞれの状況の入力モードから別の入力モードへとコマンドにより使うデバイスを適宜切り替えて入力をおこなう方法があればよい。

そこで、座っている時、立っている時、歩いている時に対して、状況に応じて変化させることのできる入力方法を提案する。

6.1 座っている時

座っている時に使うデバイスの変化を図8に示す。始めは足だけしか使えない状況を想定しており、左足で選択を行い右足の右に振るジェスチャで決定、右足の左に振るジェスチャで戻るを入力する。右に振るのは、iPod や普段のファイル操作等の階層構造では決定は右側に進むイメージがあるためである。左手が使えるときには、選択は足よりも手の方が速いので、左手を選択とする。右手が使える場合は、右手を選択とする。さらに両手が使える場合には、決定が利き手の方がよいという結果から右手を決定とし、左手を選択にする。

この遷移を自動的に行うのが望ましいが、どの手が

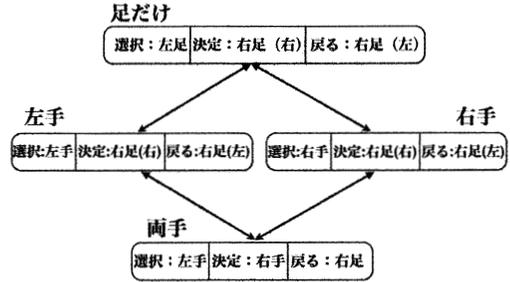


図 8 座っている時の状態変化

使えるのかを完璧に把握することは難しいため、どの手が使えて、どの手が使えないのかコマンドにより入力することとする。

左手が使える： 左手を内側外側に素早くひねる。

右手が使える： 右手を内側外側に素早くひねる。

右手が使えない： 左足で2回右方向へ振る。

左手が使えない： 左足で2回左方向へ振る。

このように使える手がある場合は、その手を使ったコマンドをおこない、使えない場合は足をつかって指示することで、使いたいときに使いたい手をつかうことができる。

6.2 立っている時

立っているときの場合は座っているときとは異なり、足によるジェスチャはどちらかの足に体重をかける必要があるために、左足と右足を同時にジェスチャ入力することが困難である。そこで、始めは片足だけを使って入力を行う。左手が使えるときには、左手を選択とし、右手が使えるときには右手を選択とする。両手が使える場合は選択と決定をそれぞれ左手と右手に割り当てる。片足だけでは常に体重が片方にかかり不可がかかるので入力する足を変えるコマンドがある。入力する足を変えると図のすべての足が逆になる。

左手が使える： 左手を内側外側に素早くひねる。

右手が使える： 右手を内側外側に素早くひねる。

右手が使えない： 使っていない足で2回右方向へ振る。

左手が使えない： 使っていない足で2回左方向へ振る。

入力する足を変える： 使っていない足を左右に振る。

6.3 歩いているとき

歩いているときには足だけでは選択にあたる動作ができないので、時間により自動的に選択肢を動かすようにする。しかし、常に選択肢が移動してしているは邪魔なときもあるため、止められるように自動選択の

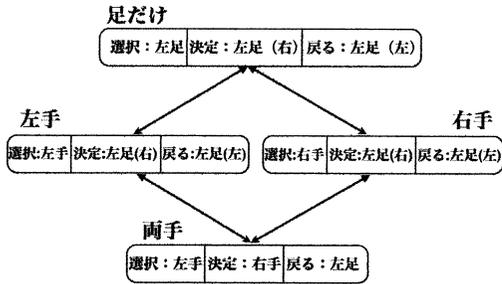


図 9 立っているときの状態変化

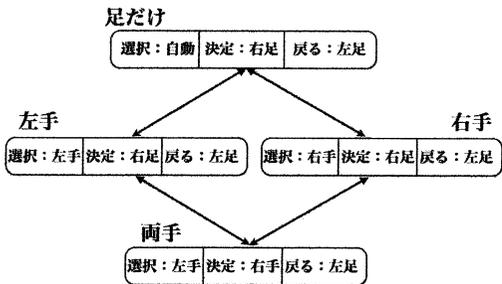


図 10 歩いているときの状態変化

on/off 用のコマンドをつくる。そのため、足を後ろに上げるジェスチャを加え、前に蹴るのと同じように DP マッチングで認識を行う。そして、右足で前に蹴る動作を決定、左足を前に蹴る動作を戻にする。左手が使えるときは左手を選択とし、右手が使えるときは右手を選択とする。両手が使える場合は、左手を選択右手を決定とする。

- 左手が使える： 左手を内側外側に素早くひねる。
- 右手が使える： 右手を内側外側に素早くひねる。
- 右手が使えない： 右足を後ろに上げる。
- 左手が使えない： 左足を後ろに上げる。
- 自動選択 on/off： どちらかの足を後ろに上げる。

6.4 考 察

上記のモード切替によって、座っている時、立っている時、歩いている時に対して、状況に応じて変化させることができる。さらに、歩いているか立っているかを加速度センサで判断できるようになれば、日常的な状態であれば、どんな状況でも入力ができるようになると思われる。

7. まとめと今後の課題

本研究では、入力デバイスには加速度センサの組み

込まれた腕輪型デバイスと靴型デバイスを用いた両手両足でのジェスチャ入力方法を実装した。実験により、両手両足の加速度センサによる入力の特徴について明らかにし、状況の変化を考慮した入力方法を提案した。今後の課題として、状況の変化を考慮したコマンド入力手法の有効性についての実験とセンサによる状況の認識が挙げられる。

謝辞 本研究の一部は、科学研究費補助金基盤 (A)(20240007, 20240009) および特定領域研究 (19024056, 19024046) によるものである。ここに記して、謝意を表す。

参 考 文 献

- 1) 庄司 武, 中村 聡央, 塚本 昌彦, 西尾 章治郎 : ウェアラブル計算環境における環境の変化を考慮した入力インタフェースの構築, MBL (モバイルコンピューティングとユビキタス通信研究会研究報告), Vol.2004, No.21, pp. 39-46,2004.
- 2) 山本哲也, 塚本昌彦, 義久智樹;日常生活における情報機器利用のための足ステップ 入力方式, 情報処理学会マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO 2007) 論文集, Vol.2007, No. 1, pp. 561 - 568, 2007.
- 3) Rekimoto J : Gesture Wrist and GesturePad: Unobtrusive Wearable Interaction Devicesk, Proc. of 5th International Symposium on Wearable Computers(ISWC2001), pp.21-27, 2001.
- 4) Paradiso J, Hsiao K and Hu E : Interactive music for Instrumented Dancing Shoes, Proc. of the 1999 International Computer Music Conference, pp. 453-456, 1999.
- 5) Starner T, Auxier J, Ashbrook D, Gandy M:The Gesture Pendant: A Self illuminating, Wearable, Infrared Computer Vision System for Home Automation Control and Medical Monitoring, Proc, of 4th International Symposium on Wearable Computers(ISWC2000), pp.87-94, 2000.
- 6) Kodama K, Fujita N, Yanagisawa Y, Terada T, Tsukamoto M:A rule-based acceleration data processing engine for small sensor node, Proc, of the 3rd international workshop on Middleware for sensor networks, pp. 1-6,2008.