

## 資源制約デバイスにおける3軸加速度センサを用いた姿勢推定手法

小澤 政博<sup>†1</sup> 川原 圭博<sup>†2</sup>  
川西 直<sup>†3</sup> 森川 博之<sup>†1,†3</sup>

我々は、近い将来携帯電話に搭載可能な小型センサを用いてユーザのおかれているコンテキストを推定し、このコンテキストに応じて、ユーザに適切な情報配信を行うコンテキスト情報基盤の構築を目指している。実用的なコンテキスト情報基盤の構築に向けては、計算資源が豊富でない携帯端末上でもリアルタイムにユーザの状況を推定できるようにすることが重要な技術課題になる。本稿では、ユーザが煩わされることなく身に着けることができるような携帯端末上においてもリアルタイムに信号を解析し、コンテキスト推定を行うための手法を示す。20MHzで動作する8bitマイクロプロセッサPIC18LF4620上で実験を行った結果、80%以上の推定精度が得られた。

### A User Posture Inference Scheme for Resource Limited Devices

MASAHIRO OZAWA,<sup>†1</sup> YOSHIHIRO KAWAHARA,<sup>†2</sup>  
NAO KAWANISHI<sup>†3</sup> and HIROYUKI MORIKAWA<sup>†1,†3</sup>

We are focusing on the development of a context-aware information platform which allows users to receive personalized information based on their context inferred by tiny sensors attached to their cellular phone. In this paper, we show a context inference scheme that realizes a user posture inference by a resource limited PIC microprocessor with one acceleration sensor. Experimental results show that the system can infer a user's posture (sitting, standing, walking, and running) with an accuracy of more than 80%.

#### 1. はじめに

ユビキタスネットワーク環境では、実空間に存在するあらゆるものがネットワークによって結ばれ、情報が流通するようになる。このいつでもどこでも何でもつながる世界におけるネットワークサービスには、ユーザの物理的な状況に応じて適切な情報を提供するコンテキストウェアネスという考え方を取り入れることが重要になる。

コンテキストウェアサービスを現実のものとするには、ユーザを取り巻くコンテキストをいかに実用的に取得可能とするかが重要な技術課題になる。一般的にはユーザコンテキストとは、ユーザの運動状態や、感情、そしてユーザを取り巻く状況(通勤中、休憩中、他)などを含む非常に幅広い解釈が可能な言葉として使われることが多い。本稿では、こうしたユーザコンテキストのうち、人の運動状態や姿勢について注目し、これらを加速度センサを具備した携帯端末を一つ身につけるだけで取得可能とすることを旨とする。人の運動状態や姿勢を推定することによりコンピュータを使って個人に対して効果的にか

つ的確に健康管理アドバイスをを行うシステムを構築するなどといったように、ユーザが常に身につけている端末上でコンテキストの推定を可能にさせることにより、きめ細やかなサービスを提供することが可能になると考えられる。

我々はこれまでも、単一の無線加速度センサと処理用PCによりユーザの運動状態や姿勢の推定を可能にする推定手法の研究を行ってきた<sup>6)</sup>。しかしながら、動作推定に必要な計算量に関しては検討を行っていなかったため、計算資源に乏しい携帯端末上でリアルタイムに推定処理を行わせることはできなかった。具体的には、加速度測定の測定間隔に比べると、計算処理にかかる時間が大きすぎるために正確な加速度計測を行えない、あるいは十分な頻度で推定処理が行えないというような問題点のためにリアルタイムに推定処理を行わせることができなかった。そこで我々は従来の推定アルゴリズムに対し、ハードリアルタイム性を考慮したタスクスケジューリング、毎回の推定計算にかかる計算時間への考慮、数値計算の高速化手法、および計算桁と推定精度のトレードオフへの考慮を加えることにより、計算資源の乏しい携帯端末上でも、新しくデバイスを追加することなく推定計算を行うことを可能にした。また、20MHzで動作する8bitマイクロプロセッサPIC18LF4620上で実験を行った結果、80%以上の推定精度を得ることができた。

本稿では、第2章において我々がやってきた従来の動作推定手法を説明する。第3章では今回計算資源に乏し

<sup>†1</sup> 東京大学 大学院新領域創成科学研究科  
Graduate School of Frontier Sciences, The University of Tokyo, The University of Tokyo  
<sup>†2</sup> 東京大学 大学院情報理工学系研究科  
Graduate School of Information Science and Technology, The University of Tokyo  
<sup>†3</sup> 東京大学 国際・産学共同研究センター  
Center of Collaborative Research, The University of Tokyo

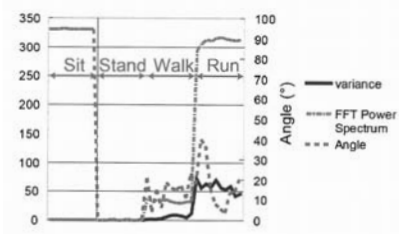


図1 ズボンのポケットの時の値の変化

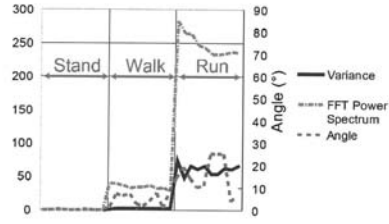


図2 上着のポケットのときの値の変化

いデバイスとして想定したデバイスについて述べる。第4章では計算資源に乏しい携帯端末上でリアルタイムに動作推定をする際に問題となる問題点を説明する。第5章では第4章で述べた問題点に対するアプローチを説明し、第6章で実験結果を示す。第7章で関連研究について述べ、最後に第7章でまとめとする。

## 2. 従来の動作推定手法

本章では我々がこれまでに行ってきた、単一の加速度センサのみを用いてセンサがズボンのポケット、胸ポケット、および鞆のちいづれに装着されている、ユーザが現在「立っている」「座っている」「歩いている」「走っている」のどの状態にあるのかを判別する手法について述べる。本手法は、推定アルゴリズムをユーザの運動状態の推定を行う部分と端末が取り付けられている位置の推定を行う部分とに分け、端末の位置の推定結果に応じてユーザの運動状態の推定に用いられるアルゴリズムを動的に切り替えるところに特徴がある。

### 2.1 ユーザの運動状態の推定

ユーザの運動状態の推定は、50msec 間隔で測定した加速度の値について、最近4個の平均値、最新12個の分散、最新64個のFFTのパワースペクトラムの最大値、重力ベクトルの方向端末の傾きの変化をあらかじめ計算し、その値を閾値に基づいて決定木を辿ることで行う。端末の傾きの変化は、一定時間前の重力加速度ベクトルと現在の重力加速度ベクトルとの内積から角度を求めている。

図1～図3は、センサがそれぞれ「ズボンのポケット」「上着のポケット」「バッグの中」にあるときの、端末の角度、分散値、FFTのパワースペクトラムの最大値を、「座っている」「立っている」「歩いている」「走っている」場合について、典型的な値の変化を示したものである。これらの図に表れている特徴を考慮し、本手法では以下のような手順によりユーザの運動状態の推定を行う。

- 分散値の大小から、動作のあるなしを推定。
- FFTのパワースペクトラムの最大値と周波数の値から、歩いているか走っているかを推定し、さらに走るリズムを推定。
- ズボンのポケットにあるとき、端末の傾きの大きな変化から座る動作を推定。

### 2.2 端末の装着位置の推定

端末の装着位置の推定は、分散値や端末の傾きの特徴的な変化を検出することで行う。

図4に端末がそれぞれのポケットにあるときの歩行時

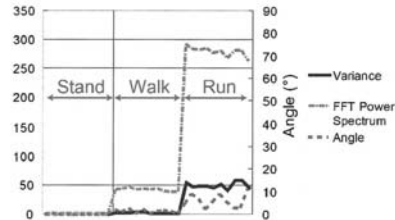


図3 バックの時の値の変化

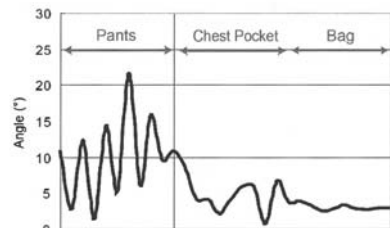


図4 歩行時の端末の傾きの変化

の端末の傾きの変化の典型的な値を示す。また、図5に端末が上着のポケットにあるときの端末の傾きの典型的な値を示す。本手法では、このような特徴を検出することにより端末の装着位置の推定を行う。以下に、本手法の従って端末の装着位置の推定の手順を示す。

- 分散値がしばらくほぼ0に近いとき、身に着けていないと推定。
- 歩いている状態のとき、端末の傾きの変化の幅が大きいと、ズボンの中と推定。
- 座る動作の瞬間、前かがみになる特徴的な動作を端末の傾きの急激な変化で検知し、その際のそれぞれの軸の平均値の変化から、端末が上着のポケットにあることと人の正面方向を推定。

以上をまとめると、図6、および図7のような推定のフローになる。

## 3. ターゲットデバイス

本研究では、ユーザコンテキストの推定処理を行わせる計算資源の制約されたデバイスとして、PAVENETモジュール<sup>4)</sup>を想定し、ユーザコンテキストの推定を

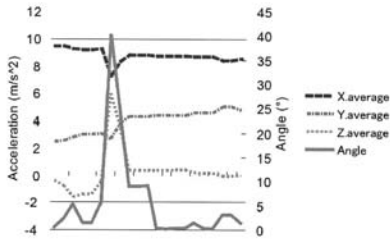


図 5 座ったときの端末の傾きの変化

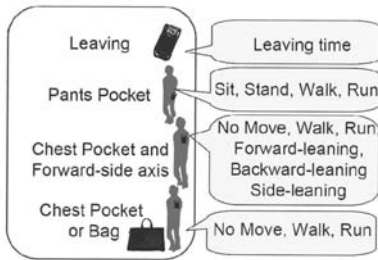
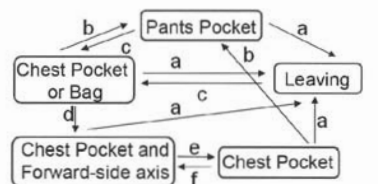


図 6 運動状態の推定



- a: The variance is nearly 0.
- b: The sensor angle fluctuate when user walks.
- c: The sensor angle don't fluctuate when user walks
- d: The sensor data shows unique change.
- e: The variance is big. f: The variance is small

図 7 端末の位置の推定

PAVENET モジュール上で動作可能とさせることを目標とした。

PAVENET モジュールは無線センサネットワーク用のテストベッドとして開発された無線センサノードである。PAVENET モジュールはマイクロプロセッサとして Microchip 社の PIC18LF4620 を具備しているが、これは電池のみで長期間動作させなければならないという無線センサネットワークの要件に向けて、計算能力よりも低消費電力性を重要視して選定されたものである。そのため、現在の PDA や携帯電話に搭載されているプロセッサと比較すると PAVENET モジュールが備えている計算資源は非常に乏しいものである。そのため、PAVENET モジュール上で動作可能な推定手法であれば、PDA や携帯電話などのデバイス上でも広く動作可能であると考えられる。

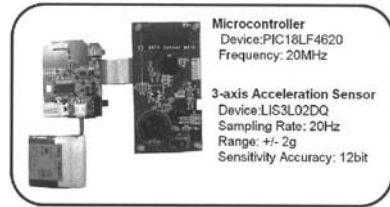


図 8 PAVENET モジュール

### 3.1 PAVENET モジュール

PAVENET モジュールは Microchip 社製のマイクロコントローラ PIC18 シリーズを搭載し、C 言語でプログラムを記述することにより、センサやアクチュエータを連係動作させることが可能である。マイクロコントローラの具体的な型番は PIC18LF4620 である。PIC18LF4620 は 8bit マイクロプロセッサであり、PAVENET モジュールでは 20MHz で駆動する。

また、PAVENET モジュールは単体では 315MHz の微弱無線で通信することができ、Bluetooth ボードを組み合わせることで Bluetooth 対応の非 PAVENET デバイスとの通信も可能になる。

### 3.2 加速度センサ

本研究の最終目標は、携帯電話のような小型端末にセンサを内蔵し、小型端末単体でコンテキスト推定をすることである。このために小型で軽量である半導体を用いた加速度センサを用いる。加速度センサは、原理の違いによりいくつかの種類があるが、本研究では、感度や精度が高く、今後さらに MEMS による高性能化が望まれるとの理由から、静電容量型の 3 軸加速度センサを選択した。具体的には ST マイクロエレクトロニクス社製の LIS3L02DQ を使用した。測定可能な範囲は  $\pm 2g$  で、分解能は 12bit となっている。

図 8 に PAVENET モジュールと PAVENET モジュールに接続した加速度センサを示す。

## 4. 従来のアルゴリズムの問題点

本章では、コンテキスト推定アルゴリズムを計算資源の乏しいデバイスで動作させる場合の問題点を明らかにする。

従来のコンテキスト推定手法をそのまま PAVENET モジュール上へ移植したところ、そのままではユーザのコンテキストを推定することはできなかった。計算速度の違いにより、PC に搭載されているような CPU 上で計算を行う場合と比較すると、PIC 上で計算を行なった場合にはコンテキスト推定のための計算時間が長くなってしまふ。そのため、PC 上で計算を行わせているときには考慮の必要がなかった問題点が顕在化してしまったためと考えられる。具体的な問題点を調べてみると、以下のような問題点があることがわかった。

- 正確な間隔での加速度測定ができていない
- 加速度の測定間隔以内に計算が終わらない場合への考慮がない

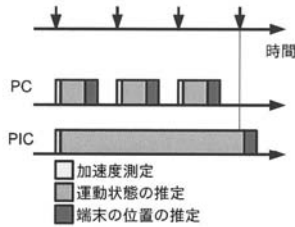


図 9 加速度測定の問題点

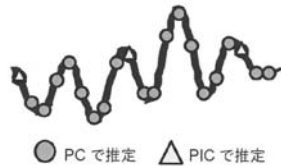


図 10 推定頻度の問題点

- 十分な頻度で推定計算ができない  
以下では、それらについて説明する。

まず、正確な間隔での加速度測定ができていないという問題点について説明する。ユーザの動作推定には測定した加速度データから求めた分散値や周波数スペクトラムを用いるため、正確な間隔で加速度を測定する必要がある。ところが、PIC上で計算を行わせたところ、図9のように、PC上で計算を行った場合は加速度の測定間隔以内に推定のための計算ができていたのに対し、PIC上で計算を行った場合には、加速度の測定間隔以内に推定のための計算を終えることができなかった。そのため、加速度の取りこぼしや測定のタイミングにずれが生じてしまい、正確な間隔での加速度測定ができていないという問題点が生じてしまっていた。

次に、加速度の測定間隔以内に計算が終わらない場合への考慮がないという問題点について説明する。従来のユーザコンテキストの推定手法では、運動状態を推定したあと、その結果に基づいて端末の位置の推定を行うというアルゴリズムになっているが、従来の手法での端末の位置推定は、毎回50msec間隔でユーザの運動状態の推定が完了するという前提のもとに端末の角度の特徴量の検出を行っている。そのため、50msec間隔で運動状態の推定が終わらないと端末の位置の推定を行うことができないという問題点が生じてしまっていた。

最後に、十分な頻度で推定計算ができないという問題点について説明する。上記で述べたとおり、従来の手法では端末の運動状態の推定結果に基づいて、端末の位置の推定を行っている。仮に、従来の推定手法に改良を施して、ユーザの運動状態の推定が50msec間隔で完了せず、疎に取得した計算結果から端末の位置の推定を可能にしたとしても、図10のように十分な頻度で運動状態の推定を行うことができなくなってしまうため、端末の位置の推定が行えなくなってしまうという問題点が生じてしまうと考えられる。

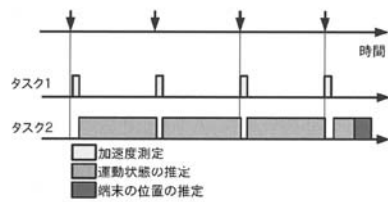


図 11 ハードリアルタイムタスクスケジューリング

## 5. 計算資源の乏しい端末上での動作推定へアプローチ

本節では、前節で挙げた問題点に対するアプローチを説明する。

まず、正確な間隔での加速度測定ができていないという問題点は、ハードリアルタイム性を考慮したタスクスケジューリングを行うことで解決する。また、加速度の測定間隔以内に運動状態の推定が終わらないために、端末の位置推定の特徴量抽出ができないという問題点に対しては、位置推定のための特徴量抽出が疎な間隔でしか取れない場合も考慮するようにして解決する。以上の2つのアルゴリズムの改善により、計算速度が遅いCPU上でもアルゴリズムが動作することの保証を行い、その上で、古典的な数値計算の高速化手法、および計算桁と推定精度のトレードオフとの考慮を加えることにより、十分な頻度で推定できないという問題点を解決する。

### 5.1 ハードリアルタイム性を考慮したタスクスケジューリング

正確な間隔での加速度測定ができないという問題点は、計算処理を行うタスクと、加速度測定を行うタスクを別スレッドに分け、加速度測定を行うタスクに関してはハードリアルタイム性を保証することによって解決する。具体的には、図11のように加速度測定を行うタスクにはより高い優先度を設定することで、運動状態の推定の計算がおこなわれている間であっても、加速度の測定を優先させることにより、正確な間隔での加速度測定を行わせる。

### 5.2 推定時間を考慮した位置推定アルゴリズム

ハードリアルタイム性を考慮したタスクスケジューリングを行うことにより、運動状態の推定は行えるようになる。

PC上であれば50msecごとに運動状態の推定を終えることができるが、PIC上では例えば200msecごとに一度しか運動状態の推定を行うことができなくなってしまうかもしれない。必ず50msecに一度運動状態の推定が終わるとは仮定せず、50msec、100msec、150msecなどに一度の頻度でしか運動状態の推定が行えない場合、すなわち端末の位置の推定に用いるデータがまばらにしか取れない場合でもそのデータで端末の位置の推定を行う用にアルゴリズムに変更を加えた。

### 5.3 古典的な数値計算の高速化手法の適用

古典的な数値計算の高速化手法として、本システムでは、三角関数のテーブル参照、および浮動小数計算の除去を行った。

ユーザの運動状態の推定には加速度の平均値、分散値、端末の角度、および周波数スペクトラムを用いるが、特に周波数スペクトラムを求める際のFFTの計算コストが大きく、そのなかでもsinやcosの値を求める演算に大きく時間がかかっていることが分かった。任意の値での三角関数の値を求めるためには、毎回テイラー展開などで値が収束するまで演算を行う必要があるが、64要素のFFTの場合を考えると、必要になる値の数は高々64点の値でしかないので、あらかじめ計算しておくことが可能である。そこで、必要なsin関数の値はあらかじめ計算してテーブルに保存しておき、実際のプログラム実行時にはテーブルから参照させることでsin関数の計算コストをおさえた。

また、浮動小数の計算はコストがかかるため途中の計算はすべて整数として計算を行った。値が1~1の範囲となる三角関数については、あらかじめ定数倍したものの整数部分だけを利用し、三角関数を用いた計算をするごとにその定数で割ることにより浮動小数の計算の除外を行った。

#### 5.4 計算桁と推定精度のトレードオフへの考慮

PICは演算のビット幅が8bitである。そのため、16bitや32bitの計算はそれだけ計算コストがかかってしまうと考えられる。

計算の桁数を落とせばそれだけ高速に計算を行うことが可能になると考えられるが、桁数が少なくなるために推定精度が落ちてしまうと考えられる。

そのため、推定の精度を落とさない範囲で、なるべく計算に用いるビット幅を少なくするようにした。

## 6. コンテキスト推定精度の評価

### 6.1 計算時間

PAVENETモジュール上で推定計算を行わせたときの、計算桁、計算時間に関して実験を行った。

最低限必要な計算桁を実験によって確認した。まず、計算に用いる桁数を8bitとして推定のための計算を行わせてみたところ、FFTに関してまったく意味のある値がとれなかった。次に、計算に用いる桁数を16bitとして推定のための計算を行わせてみたところ、ほぼ正しい結果が得られることがわかり、16bitで推定を行わせることにした。

計算桁が32bit、三角関数のテーブル参照の考慮なし、および浮動小数計算の除外をしていない場合、毎回のユーザの運動状態の推定にかかる時間はおよそ3secであった。このとき、端末の位置の推定を行うことはできなかった。計算桁が16bit、三角関数のテーブル参照、および浮動小数計算の除外をしている場合、毎回のユーザの運動状態の推定にかかる時間はおよそ170msecであった。

次節以降の推定精度の結果は、計算に用いる桁数を16bitとしたときの値である。

### 6.2 動作の認識の精度

それぞれの装着場所における動作の認識について精度を評価した。4人の被験者に対して、座る(ズボンのときのみ)、立つ、歩く、走る動作を10秒ずつおよそ2分間繰り返すことを、ズボンのポケット、上着のポケット、

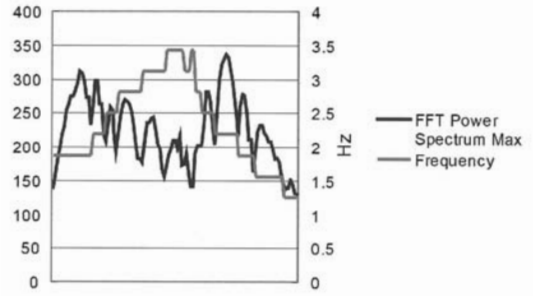


図 12 走るリズムの相対的な変化

鞆の中の3箇所それぞれ行った。被験者は、20代男性4人である。実験前に被験者が脚に異常を感じていたり、怪我をしていないことを確認した。実験は自由に動き回れるように十分な広い空間で行った。ズボンはそれぞれの被験者が普段はいているものを使用し、後ろポケットや横ポケットの選択はユーザの判断に任せた。上着のポケットは、Yシャツを着ていた被験者はそのポケットを、ポケットのついていない服を着ていないユーザはアウターのコートについているポケットを使用した。鞆は、それぞれの被験者が普段使っているハンドバッグ、トートバッグ、リュックサックを使用した。端末のコンテキストと動作推定の正解率を求めた結果を表1に示す。端末の装着場所と動作の推定を高い精度で実現できた。

表 1 動作の推定精度

	sitting	standing	walking	running	average
Pants	99.2%(129/130)	80.2%(109/135)	92.5%(124/134)	98.5%(133/135)	92.7%(655/704)
Bag	99.4%(159/160)	99.5%(154/155)	99.8%(159/160)	99.3%(171/173)	
Chest	99.4%(159/160)	95.0%(152/160)	98.1%(157/160)	98.3%(668/680)	

### 6.3 走るリズムの精度の推定

FFTで走るリズムを正確に推定できているか、実験を行った。相対的な変化をとらえることができているかをまず調べた。図12に走るリズムを意図的に段階的に早くし、その後また段階的に遅くして、その変化を観測した結果を示す。1.8Hzから3.5Hzまで上昇し、その後下降していることが確認できる。これより、計算に用いる計算桁を16bitにした場合でも、走るリズムの相対的な変化を推定できていることがわかる。

## 7. 関連研究

人の運動状態や姿勢をコンピュータを用いて推定しようとする場合、ビデオカメラなどを環境側に用意し画像処理により推定する方法と、加速度センサをユーザの身体に装着する方法に大別できる。前者はユーザに何も装着する必要がないという点がメリットであり、後者は環境側に特殊な装置を用意する必要がないというのがメリットである。本研究では場所を特定せずモバイル環境での実現を目指すため後者のアプローチについて検討を行う。

以降では、身体の各部位の加速度の値の大小から日常

生活における動きを推定しようとする研究に関して、加速度センサからの信号をどこで処理するのかという観点から関連研究を大きく3つにわけ、それぞれのシステム構成、推定項目、目標としているサービスについて紹介する。

#### 7.1 処理用のPC上で推定を行う関連研究

Carnegie Mellon 大学の SenSay では、携帯電話に複数のセンサ情報を加えることで、「取り込み中」「活動中」「暇」「ノーマル」の4つの状態に分類して、着信音の音量をサイレントモードにするアプリケーションなどを実装している<sup>5)</sup>。ノイズを感知するためのマイク、3軸の加速度センサ、可視光センサ、温度センサから得るデータを利用している。収集されたデータは処理用のPCへ送られ、PC上で処理を行っている。

ETH Zurich の Kern らは肩、肘、手首、腰、膝、足首に合計12個の3軸加速度センサを装着し、Bayesian 推定を用いて「座っている」「立っている」「歩いている」「階段を上る」「階段を下る」「握手」「黒板への書き込み」「キーボードへのタイピング」を推定している<sup>2)</sup>。また、このシステムを利用した、煩わしさの推定に取り組んでいる<sup>3)</sup>。

MIT の Stephen らは両手首、両足首、腿の合計5箇所にも2軸加速度センサを装着し、動作推定を試みている<sup>7)</sup>。およそ6.7秒分のデータから平均、エネルギー、周波数領域エントロピー、そして相関を求め、C4.5の分類法で推定している。推定する動作は20種類あり、関連研究の中では最も多い。また、個々のユーザで閾値を設定せずに推定できる汎用性を狙っているのが特徴として挙げられる。

#### 7.2 サーバー上で推定を行う関連研究

NTT DoCoMo の Iso らは、携帯電話に複数のセンサを搭載したスマートフォンの研究をしている<sup>1)</sup>。3軸加速度センサ、3軸ジャイロセンサ、マイク、赤外線センサ、明るさセンサ、温度センサ、圧力センサ、皮膚感知センサ、血圧センサを利用している。ユーザの動作についてのコンテキストは、3軸加速度センサと3軸ジャイロセンサを用いて、「歩く」「走る」「座る」を推定している。センサから集められたデータは一度ネットワーク上のサーバーへ送られ、サーバー上で処理される。

#### 7.3 省資源デバイス上で推定を行う関連研究

マイクロストーン社の市川らは、腕時計型運動センサを研究している<sup>7)</sup>。推定処理は腕時計の中で行われ、一軸加速度センサと一軸ジャイロセンサの値の大小、周期性のあるなしから「散歩」「平常歩」「速歩」「急歩」「ジョギング」「ランニング」「座位作業」「立位作業」「軽いスポーツ」「激しいスポーツ」の10種類に運動パターンを識別する。それぞれの動作に応じた係数を用いて消費カロリーを算出する。

## 8. おわりに

本稿では、計算資源の制約された携帯端末上でリアルタイムにユーザの運動状態や姿勢を高い精度で推定するコンテキスト推定手法について述べた。具体的には、従来我々が研究を行ってきた推定手法について、ハードリアル

タイム性を考慮したタスクスケジューリング、運動推定の計算時間の考慮した端末の位置推定のための特徴量抽出、古典的な数値計算の高速化手法の適用、計算桁と推定精度のトレードオフへの考慮を加えることにより、計算資源の制約された携帯端末上でもリアルタイムにユーザのコンテキストを推定することが可能になった。実験の結果、20MHz駆動の8bitマイクロプロセッサPIC18LF4620上で、80%以上の推定精度を得ることができた。

## 参考文献

- 1) Toshiki Iso, Norihiro Kawasaki, and Shoji Kurakake. Personal context extractor with multiple sensor on a cell phone. In *The 7th IFIP International Conference on Mobile and Wireless Communications Networks, D. 2 C200525*, Morocco, September 2005.
- 2) Nicky Kern, Stavros Antifakos, Bernt Schiele, and Adrian Schwaninger. A model for human interruptibility: Experimental evaluation and automatic estimation from wearable sensors. In *Proceeding of 8th IEEE International Symposium on Wearable Computers (ISWC'04)*, pp. 158 - 165, Arlington, U.S.A, October 2004.
- 3) Nicky Kern, Bernt Schiele, and Albrecht Schmidt. Recognizing context for annotating a live life recording. *Personal Ubiquitous Comput.*, Vol.11, No.4, pp. 251-263, 2007.
- 4) Shunsuke Saruwatari, Takuya Kashima, Masateru Minami, Hiroyuki Morikawa, and Tomonori Aoyama. Pavenet: Hardware and software framework for wireless sensor networks. *Transaction of the Society of Instrument and Control Engineers*, Vol. E-3, , 2005.
- 5) Daniel Siewiorek, Asim Smailagic, Junichi Furukawa, Neema Moraveji, Kathryn Reiger, and Jeremy Shaffer. Sensay: A context-aware mobile phone, October 2003. Poster of 7th IEEE International Symposium on Wearable Computers (ISWC'03).
- 6) 倉沢央, 川原圭博, 森川博之, 青山友紀. センサ装着場所を考慮した3軸加速度センサを用いた姿勢推定手法. 情報処理学会研究報告, ユビキタスコンピューティングシステム研究会 (UBI-11-3), May 2006.
- 7) 市川和豊, 井澤裕司, 北澤俊二, 芳川美代子, 浜淳. 426 人体の行動識別が可能な腕時計型装置の開発. パイオエンジニアリング講演会講演論文集, 2003(15), pp325-326, 2003.