

平成 27 年度博士前期課程(ソフトウェア情報学)論文

振動センサを用いた人間の行動推定システム
の開発

A system to recognize human behavior
using vibration sensors

岩手県立大学大学院
ソフトウェア情報学研究科

学籍番号：2312014005

内田 泰広

研究指導教官：澤本 潤

杉野 栄二

要旨

高齢化社会に伴い独居老人の人口が増え高齢者の孤独死も発生し社会問題となりつつある。このような問題を背景に様々な見守りシステムが開発されてきた。

しかし先行研究の多くは人間に装着する方式の見守りシステムであり高齢者にとっては負担が大きいという問題点がある。また上記の方式の場合、センサを装着するのを忘れる場合や人間が決まったイベントを起こさない限りは行動情報を観測できないという問題があった。

先行研究では床面から得られる振動情報を観測することで高齢者に負担をかけずに人間の行動情報を推定できる手法について開発してきた。

このシステムは、高精度な加速度センサを床に設置し、人間が歩いた時などの振動情報をパソコンに取り込むことで一定時間の行動推定を行った。実装では短時間（55 秒）の行動推定しか行えないため、随時高齢者の行動を観測し続けるためのシステムが必要になる。

そこで本研究では、先行研究で実装していた振動解析システムを従来の PC から、Raspberry Pi とタブレット端末上で実装することにより、高齢者をリアルタイムに観測できる見守りシステムを構築する。また実装上の工夫により複数センサを Android 端末 1 台で一元的に処理できる方法の実現、低コスト化をできたことについて報告する。

本論文で提案するシステムは、主に振動を観測するセンサ部（入力）、振動データから行動を推定する受信処理部（解析）の 2 つのモジュールで出来ている。センサ部には Raspberry Pi、受信処理部には Android 端末を使用して実装を進めた。Raspberry Pi には、高精度加速度センサと Bluetooth モジュールが接続されており、Bluetooth モジュールは加速度センサが床面から取得した振動情報を Android 端末に向けて送信する。

Android 端末で Bluetooth 通信を正常に行うためには様々な制約がある。従来(Google Play で配布されている)アプリケーションではバッファからデータを読み込むタイミングが遅い場合や、受信したデータを適切に処理していないため、データを取りこぼすなどの問題が多発していた。これらを解決するため、メソッドを呼び出すタイミングを短くし、受信後に適切な処理をすることでこの問題を克服した。

実験では、部屋に数台のセンサを設置し歩行などの行動の予測や位置の推定も行うことができることを確認した。これにより先行研究では行えなかった、リアルタイムに人間の行動を同定するシステムの実装が出来たことを報告する。

Abstract

For aged society. Death of living alone in senior citizen's have been increasing. Because of these problems, various watching system have been developed.

Most of the existing research, old man feel to bear wearable device. And, problems occurs that forget to wear the device. When man doesn't do decided behavior, behavior can't be detected.

Previous research has developed the way at which human behavior information can be estimated from shaking on a floor. This system is connecting a highly precise acceleration sensor(Raspberry Pi) with a PC, and a sensor detects shaking when a man walked. I need the system always to observe senior citizen's behavior because short time's behavior can be presumed at present.

So by this research, The system to observe vibration is programed in a tablet and the system to watch a senior citizen in real time is built.

By my research, Realization of way which can manage more than one sensor unitarily by a tablet and achievement of cost reduction will be reported.

Systems are made of 2 modules. Sensor module and processing Module. Sensor module made of Raspberry Pi and acceleration sensors. Processing module made of Android device. A highly precise acceleration sensor and Bluetooth module are connected to Raspberry Pi. Bluetooth module sends vibration information for a Android terminal. By Android device Bluetooth communication, there are various rules normally to communicate. The timing by which data is read from a buffer is slow by application of existence. To lose data occurs because I don't deal with reception data right.

To settle these problems. The timing to which the method is called, a short comb. And a problem was settled because it was received right. By the experiments, it was confirmed that enables estimation of human behavior and location.

Therefore, we will report that the system for estimating the human behavior using the vibration information in real time could be created.

目次

1 目次

1. 研究背景	1
2. 見守りシステムに関連する研究	3
2.1. 据え置き型振動センサを用いた人の行動同定	3
2.2. 複数センサを用いた人間の生活行動データ抽出システム	6
2.3. 装着型センサ（スマートフォン内蔵センサ）を用いた行動判定	6
2.4. 見守りシステムに関連する研究のまとめ	7
3. 加速度センサを用いた人間の行動推定システムのプラットフォーム	8
3.1. 研究目的	8
3.2. システム全体の流れ	8
3.3. センサ部	10
3.4. 加速度センサのデータフォーマット	11
3.5. センサ取得から加速度を算出する式	12
3.6. Android アプリケーション部	13
3.6.1. Android 端末を選定した理由	13
3.6.2. Android アプリケーション部の処理（全体）	13
3.6.3. Android アプリケーション部の処理（受信処理）	16
3.6.3.1. 受信バッファのサイズ・読み込みスレッド待機時間の問題	16
3.6.3.2. Byte 型から String 型へのキャスト変換問題	17
3.6.4. Android アプリケーション部の処理（複数センサから同時受信） ..	18
3.6.4.1. 受信処理（データ処理スレッド）	20
4. 評価実験	21
4.1. 実験の目的	21

4.2. 実験 1 : 1 台のセンサと Android 端末の Bluetooth 通信実験	21
4.2.1. 実験 1 の考察	23
4.3. 実験 2 : 複数台のセンサと Android 端末の Bluetooth 通信実験	24
4.3.1. 実験 2 の考察	24
4.4. 実験 3 : 排他制御の実装による遅延の計測	25
5. おわりに	26

図目次

図 1 ネットワークカメラ (TS-WLCAM/V) [5]	1
図 2 振動波形・スペクトル強度 [6]	2
図 3 行動同定システムのフロー	3
図 4 行動同定システム	4
図 5 据え置き型振動センサの設置	5
図 6 システム全体の流れ	8
図 7 システムの処理フロー	9
図 8 センサモジュール	10
図 9 センサのスイッチ (ON/OFF)	10
図 10 加速度センサデータフォーマット	11
図 11 加速度を算出する式	12
図 12 Android アプリケーション内の構成	14
図 13 Android アプリケーションのシーケンス	15
図 14 中間バッファから振動データを読み込む処理	16
図 15 読み込み動作	17
図 16 バッファサイズの調整	18
図 18 Bluetooth のマスター・スレーブ関係 (ピコネット)	18
図 19 センサ ID を使った受信処理	19
図 20 複数の受信スレッドとメッセージ処理受信	19
図 21 ダブルバッファリング	20
図 22 安静時の振動データ(センサ 1 台)	21
図 22 歩行時の振動データ(センサ 1 台)	22
図 23 安静時の振動データの周波数解析(センサ 1 台)	22
図 24 歩行時の振動データの周波数解析(センサ 1 台)	23
図 25 歩行時の振動データの周波数解析(センサ複数台)	24

表目次

表 1 Android アプリケーションで利用できるセンサ	6
表 2 データフォーマットの詳細	12

1. 研究背景

近年、少子高齢化社会が進んでおり社会問題の一つとしても注目されてきている。総務省統計局が行った調査によると平成 25 年 9 月時点で国民の約 4 人に 1 人が高齢者であり、20 年後には 3 人に 1 人が高齢者という超高齢化社会になると予想されている[1]。また高齢者にまつわる事故や問題も増加してきている。高齢者にまつわる社会問題として独居老人の孤独死があげられる。普段連絡を常にしあう家族や近所の人などによる定期的な声掛けなどがなく、社会から孤立してしまう人に孤独死が発生してしまう。これらの問題を解決するためには定期的に高齢者の行動や異常検知が必要である。人間が高齢者の様子を随時監視することは不可能であり、周囲の負担も大きい。そのため世の中では様々な見守りシステムが開発されてきている。見守りシステムでは高齢者に負担の少ないように主に 3 つのことを重視している。

1 つ目はプライバシーの確保である。ネットワークカメラは見守る側としては一番簡単な方法であり一番安心感のある方法であるが、見守られる側（高齢者）に取っては常に行動を映像として記録され続けることに不安を感じる。品川らが行った研究によると[2]、高齢者は機械に見守られることに対する安心感と同時に不安感も同様に持っていることが分かった。また、高齢者はカメラでの撮影には大きな抵抗感があるが家の何処にいるかを特定されるなどに対してはさほど抵抗感がないことが判明した。高齢者に抵抗感が少ない見守りシステムとして姿などを撮影しない重要性を明らかにし、プライバシーを考慮した見守りの方法を提案している。高塚らの研究ではカメラを使うことは見守りでは監視行為であると議論になりプロジェクトが中止となる事態が発生した[3]。高齢者をただ見守るという目的を果たす以上に見守られる側の感情も考慮したシステムの開発が求められてきている。



図 1 ネットワークカメラ (TS-WLCAM/V) [5]

2つ目はコストである。高齢者の過半数は年金で暮らしている[4]。そのため、高価な安否確認サービスを受けるために高価な機材で見守りをするのは困難である。出来るだけ安価な手法や機材を導入することで初期投資を減らすことが重要となる。また、継続的な観測を考慮し、維持費や消費電力などの電気代のかからない方法を選択する必要がある。

3つ目は定期的な行動観測である。高齢者の安否や安全性を見守るためには定期的に行動を観測し続けることは必須である。電話オペレータによる安否確認サービスなどでは週に1～7回の電話でその日の様子や安否を確認しているが、異常検知などの発見をすることは困難であると考えられる。出来るだけ短い周期で行動を観測することで異常事態が起きた時にすぐに対応することが可能になる。観測する周期が短すぎるとデータが膨大になったりすることがあるので適度な時間で観測することが必要と考えられる。

独居の高齢者の見守りをするためには上記のプライバシーの確保、コスト、定期的な観測の項目を満たすシステム作りが理想である。

本研究では、据え置き型振動センサを使用することで振動情報から得られる振動波形の揺らぎを使うことにより、定期的な高齢者の行動を観測できる手法を提案する。本研究で使用する据え置き型振動センサは、床に3軸加速度センサを起き床に対して垂直方向（Z軸）の振動を観測するものである。

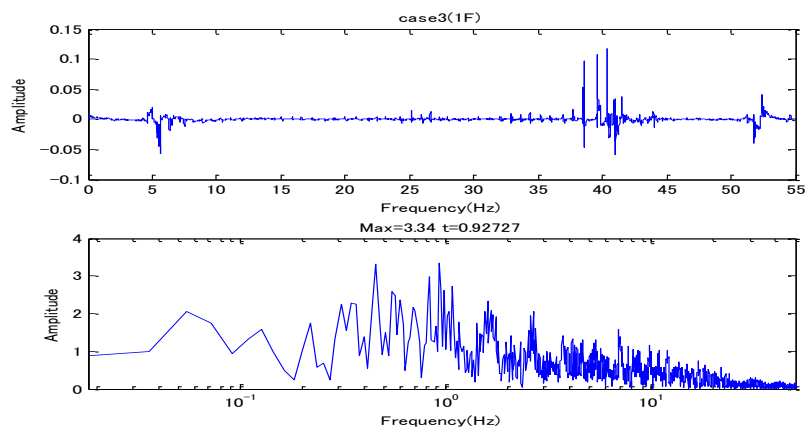


図 2 振動波形・スペクトル強度 [6]

2. 見守りシステムに関連する研究

本章では、本研究の先行研究に当たる据え置き型振動センサを用いた人間の行動推定システムの技術・アルゴリズムについての説明とその他現在行われている見守りシステムについての詳解を行う。

2.1. 据え置き型振動センサを用いた人の行動同定

情報技術を使用して高齢者を見守るためには対象者へ負担が少ないようにシステムを構築する必要がある[7]。そのような要望を解決するために据え置き型振動センサを用いることで高齢者などに装着の手間がいらずに行動のログが取れるシステムがある[6]。このシステムは独居老人を対象として人間の行動情報の同定を行っている。

建物内の床に振動観測センサとなる 3 軸加速度センサを設置し、加速度センサからは床（垂直方向）の振動を観測する。床の振動は主に建物固有の振動と人間が生活する上で発生する人工的な振動が観測される。その振動情報から微妙な揺らぎを観測することで人間の行動同定を行っている。以下に独居老人の行動推定を行うシステムの全体的な流れを示す（図 3）。

1. 床に設置された加速度センサで振動情報の観測を行い保存する（5 5 秒間）。
2. 保存した振動情報を行動同定システムに取り込む
3. 解析した行動を観測する

5 5 秒間の振動情報を同定して得られる情報は歩行中や静止中などの一つの行動（イベント）である。人間の行動が約 1 分程度で移り変わると判断したため 5 5 秒間の振動情報を使用している。加速度センサは 1 秒間に 1 0 0 サンプルの振動データを観測している。5 5 秒間分のデータは 5 5 0 0 サンプルを示している。

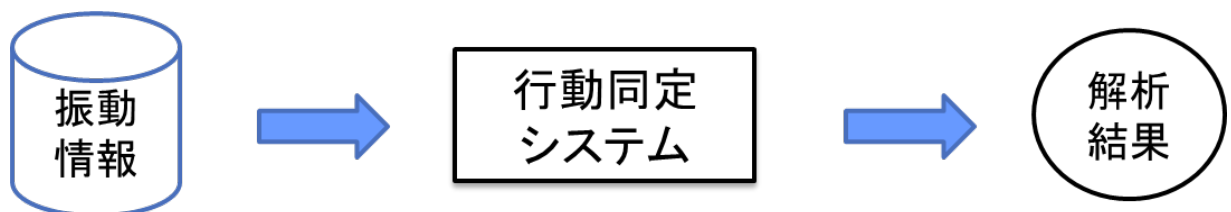


図 3 行動同定システムのフロー

行動同定システムは以下の 4 つの手法を総合的に判断し振動情報から人間の行動同定を行っている。

- 時刻歴振動波形
- スペクトル強度
- 時刻歴の頻度分布（ヒストグラム）
- スペクトル強度の頻度分布（ヒストグラム）

時刻歴振動波形は、時間軸における振動の大きさ（振幅）を見ている。スペクトル強度は振動情報をフーリエ解析することで周波数ごとのスペクトルを求めている。頻度分布（ヒストグラム）は前述のデータの頻度を求めている。振動情報からこれらの手法を行い、総合的に判断し行動判定を行っている。

図 4 に上記の行動同定するために必要な手法に関わる波形情報を掲載する。

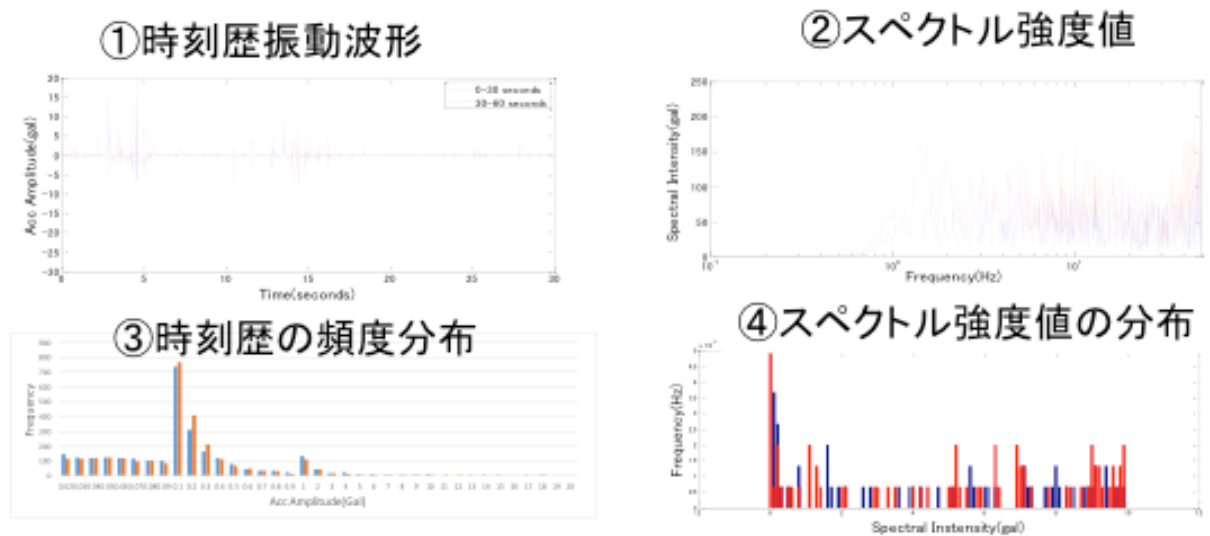


図 4 行動同定システム

このシステムの問題点は 55 秒間しか人間の行動情報を判断することが出来ないことである。1 度に 55 秒間の振動データを独立で PC に貯めこむが行動同定に取り込むシステムを作っていないため、リアルタイムで見守りができない。



図 5 据え置き型振動センサの設置

2.2. 複数センサを用いた人間の生活行動データ抽出システム

近年、家庭でのセンサネットワークを使った取り組みとして HEMS(Home Energy Management System)や、家電製品を制御する技術 (IoT など)、スマートメータなどの取り組みが頻繁に行われるようになってきている。上田らはスマートホームに複数のセンシングデバイスを設置し、屋内における人間の生活行動データを抽出する研究を行い、SVM (サポートベクタマシン) を利用することで 85.05%の適合率で行動を識別するシステムを開発した [8]。システム要件として「(1)他種類かつ抽象的な生活行動を認識できる (2)低コスト、少数センサで実現できる (3)居住者のプライバシーを侵害しない」を掲げており、複数のセンサを用いることで以上の要件を満たすことを考えている。使用しているセンサは、環境センサ (温・湿度度、照度、人感)、超音波センサ、消費電力センサなどを使用している。この研究では被験者の位置情報は装着型超音波センサを使用しており、被験者にセンサを装着しなければならないという手間が欠点となる。

2.3. 装着型センサ (スマートフォン内蔵センサ) を用いた行動判定

既存の見守りシステムのほとんどが人間の体にセンサを取り付ける手法がほとんどであるが、本研究では高齢者にかかる負担の軽減を要件として研究を行っているため、本章では非装着型で携帯型のセンサとしてスマートフォンを利用した見守りシステムについて紹介する。

現在のスマートフォン(Android 端末)には様々なセンサが搭載され、センサ情報は Android アプリケーションとして実装することで取得することできる。表 1 に Android アプリケーションで取得できる代表的なセンサのパラメータを記載する。

表 1 Android アプリケーションで使用できるセンサ

センサの種類	定数
加速度センサ	TYPE_ACCELEROMETER
地磁気センサ	TYPE_MAGNETIC_FIELD
傾きセンサ	TYPE_ORIENTATION
近接センサ	TYPE_PROXIMITY
圧力センサ	TYPE_PRESSURE
温度センサ	TYPE_TEMPERATURE
照度センサ	TYPE_LIGHT
ジャイロセンサ	TYPE_GYROSCOPE

表 1 から分かるように Android 端末から様々なセンサの値を取得することが出来る。これより被験者のログ情報を取得するのに十分な情報が取得できることが想像できる。

太田らは被験者がスマートフォンを携帯していることを前提として Android 端末内蔵の加速度センサ・GPS を利用して、振動の振幅を閾値で判断することで被験者の歩数を判定し、転倒したことを検知する[9]。GPS センサは、現在何処にいるかなどの判断を行っている。実験では高齢者の異常状態と正常状態を観測できたことを示した。しかし課題としてはアプリケーションを常に実行しセンサの情報を取得しているためバッテリーの消費が激しいこと、端末の大きさが高齢者に不向きではない、行動特定アルゴリズムが端末の影響を受けてしまうなどの課題点をあげていた。いずれの課題点も端末の制約に依存する形になってしまったことが分かった。

大野らは TLIFES というプロジェクトでは、スマートフォンを用いた弱者見守りシステムを作成している[10]。前者の研究と同じく Android 端末に内蔵されている加速度センサや GPS センサを中心に被験者のログ情報や異常を検出している。主に加速度センサを利用し振幅の回数や大きさによって異常検知や歩数の判定を行っている。文献 9,10 の研究もセンサ内蔵端末を携帯することを想定した研究である。

2.4. 見守りシステムに関連する研究のまとめ

本章では本研究の先行研究であり据え置き型振動センサを作成するために必要となる技術（行動解析アルゴリズム）を説明した。据え置き型振動センサを利用することで高齢者に負担をかけずに人間の行動を解析できることができる。しかし現在の行動解析アルゴリズムでは一定の時間しか人間の行動情報を判断することが出来ないため、長時間での見守りが不可能である。

そこで本研究では、振動情報を自動で順次解析システムに取り組むためのプラットフォーム作りを完成させ、長期的な老人の見守りが可能となるシステムを作成させる。

複数センサを用いた見守りシステムは、自宅がスマートホームのようにたくさんのセンサを設置しなければならないという点でコストが掛かってしまう為、独居老人が導入するためには金銭的に困難である。

また携帯型のセンサを用いた見守りシステム・生活行動ログシステムの場合、高齢者が常に携帯電話を持ち歩かなければならないという制約があるため、高齢者への負担や持ち忘れの発生を考慮すると導入が困難である。

3. 加速度センサを用いた人間の行動推定システムのプラットフォーム

本章では、据え置き型振動センサを用いた人の行動同定のアルゴリズムを用いて、独居高齢者の見守りシステムを構築したことについて述べる。

3.1. 研究目的

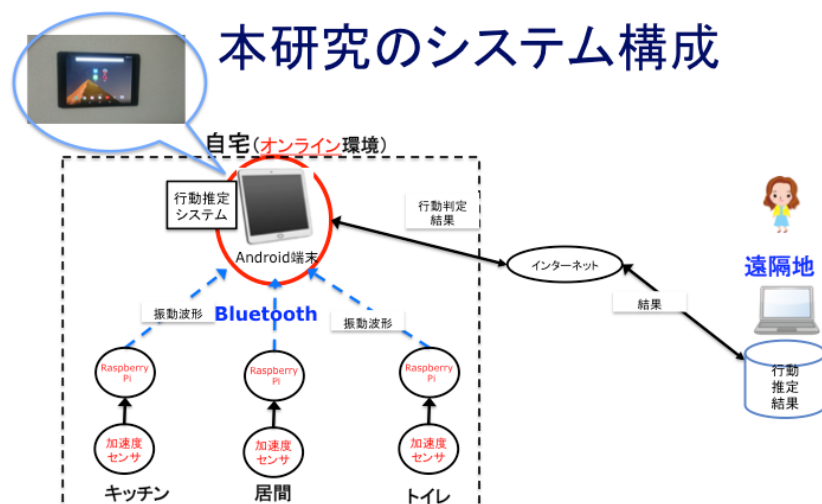
本研究の目的は以下の通りである。

1. リアルタイムで行動推定をし、見守る側が行動推定できる
2. 先行研究のシステム構成を見直すことでコストの低減[7]

これらの要件を満たすことを条件に見守りシステム構築していく。

3.2. システム全体の流れ

本研究の見守りシステムは、既存研究の据え置き型振動センサを用いた行動同定を使用し、リアルタイムに高齢者の見守りを行う。システム全体の流れを図 6 に示す。



6

図 6 システム全体の流れ

システムの処理は、以下の通りである(図 7)。

1. 床などに設置された加速された加速度センサで振動情報を観測
2. センサで観測した振動情報を無線で Android 端末に送信
3. Android 端末は振動データの受信処理を行い、振動データを取り込む
4. 55 秒間の振動データを貯めこんだら、行動同定を行う
5. 解析した情報を遠隔地のサーバに送信する

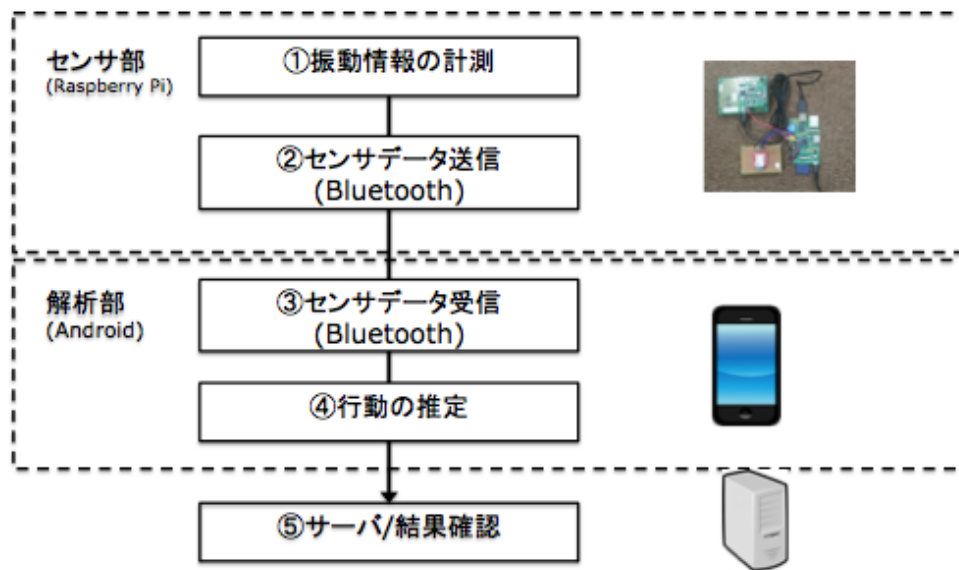


図 7 システムの処理フロー

3.3. センサ部

センサ部の構成は主に 3 つのモジュールで構成されている(図 8)。

- 高精度加速度センサ(KX-R94)
- Raspberry Pi
- Bluetooth モジュール(RN-42)[11]

センサデータの送受信はスイッチで切り替える (図 9)

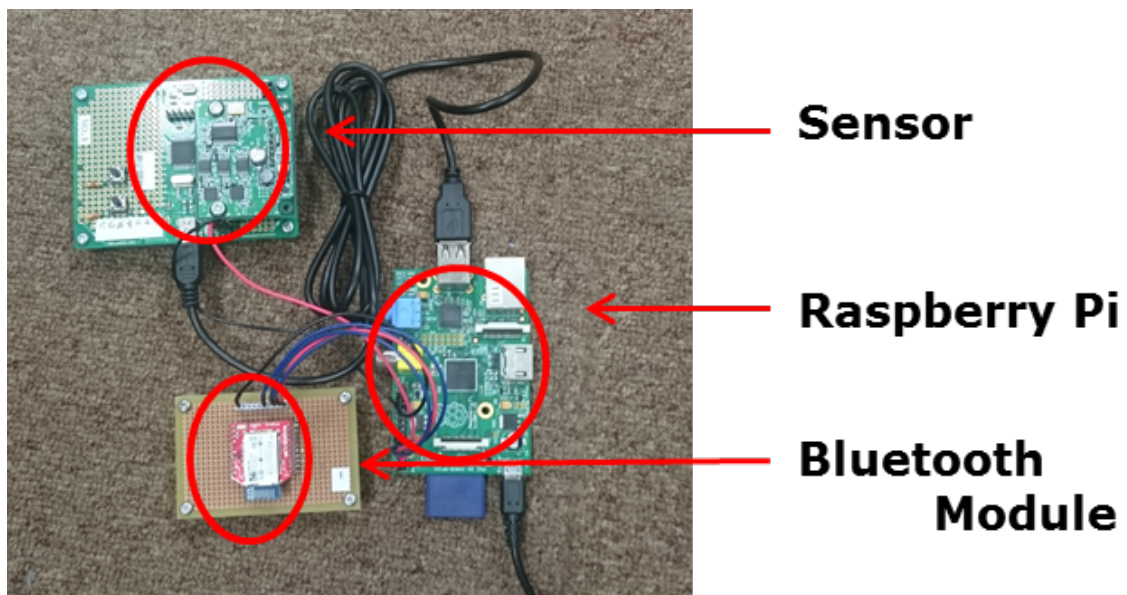


図 8 センサモジュール



図 9 センサのスイッチ (ON/OFF)

加速度センサは振動データの入力に当たるモジュールになる。床に設置し 3 軸の加速度を取得する。取得する周期は毎秒 100 サンプルである。本研究では 60 秒間のデータを蓄積したうえで、行動を解析するため、100 サンプル×60 秒の計 6000 サンプルのデータを 1 度に使用する。

Raspberry Pi は制御に当たる部分になる。Raspberry Pi 上でプログラムを動作させ加速度センサで計測した振動情報を Bluetooth モジュールで出力（送信）する。Bluetooth モジュールは Android 端末に向けてデータを送信する。現在使用している Bluetooth モジュールは Bluetooth クラス 2 を使用しているため理論上通信距離が 30m 飛ぶため、自宅でのセンサのカバレッジの問題を考慮しなくてもよい。通信レートは Bluetooth プロファイル SPP というシリアル通信を使用しているため 240kbps である。

センサモジュールに付与してあるスイッチによる Bluetooth で送受信をするか否かを帰ることが出来る。

3.4. 加速度センサのデータフォーマット

加速度センサは、3 軸加速度センサを使用しているため X、Y、Z の 3 種類のデータを取得している (図 10)。図 10 のデータフォーマットは Raspberry Pi から Bluetooth 通信を通して Android 端末に送信されるフォーマットである。センサフォーマットの詳細については表 2 に示す。

センサから取得する X、Y、Z 軸のデータはセンサ固有の値であるため、実際の行動判定に使う際は加速度[ガル：gal]に変換する必要がある。

S 001 , 04175436 , 04289491 , 05938584 E

図 10 加速度センサデータフォーマット

表 2 データフォーマットの詳細

記号 or 数字	意味
S	開始記号
001	●秒目のサンプル
04175436	X軸データ
04289491	Y軸データ
05938584	Z軸データ(垂直方向)
E	終端記号

3.5. センサ取得から加速度を算出する式

センサが取得した振動は出力 AD 値で出力される。この値を図 11 で示す式に代入することで加速度[gal]に変換できる。本研究で使用する値は垂直方向の振動なので Z 軸のデータを使用する。Z 軸は重力加速度を考慮する。重力加速度は $1G = 981[\text{gal}]$ である。1gal は 1 秒に $1[\text{cm/s}]$ の加速度の大きさとして定義される ($1\text{gal} = 0.01\text{m/s}^2$)。

センサから取得した出力 AD 値が「05938584」の場合、約 1019[gal]が算出される。この値を元に行動推定がされる。

①入力電圧の計算

- $V_{in} = 2.5 \times \text{出力AD値} / 4194304$

②加速度に変換

- $\text{加速度} = (V_{in} - 2.5\text{V}) \times 981$

図 11 加速度を算出する式

3.6. Android アプリケーション部

振動情報を集約するノードとして Android 端末を選定した理由と Android アプリケーション上での受信処理、複数センサからデータを受信する際に考慮した手法について記載する。

3.6.1. Android 端末を選定した理由

Android 端末を選定した理由を以下に述べる。

- Android 端末の低価格化
- スマートフォン内臓の複数センサとの統合
- 通信機能（Wi-Fi・Bluetooth）が備わっている

現在、中古のスマートフォンや海外の安いスマートフォン・タブレット・スマートウォッチが販売されている。これらを利用すれば低価格化でシステムの導入も可能である。また個人でスマートフォンやタブレットを持っていた場合、そのまま端末をシステムに流用することが可能である。

ログ情報を計測してサーバなどにデータを送信するとき回線契約をしなければならない。有線の場合、自宅に回線を引くとなるとどうしても高額になってしまう。MVNO 事業者の SIM カードを使用することで通信料も安く出来ることが可能である。

Android 端末には様々なセンサが搭載されている。これらのセンサを別のアプリケーションとして実装しログを取得すればより詳細な情報を得て見守りシステムが可能である。

Wi-Fi、Bluetooth を使うことで Raspberry Pi との通信、遠隔地のサーバに対するデータの送信が可能である。

3.6.2. Android アプリケーション部の処理（全体）

Android アプリケーション部は複数のセンサが床から観測した振動情報を集約し、行動解析モジュールに読み込ませる役割を持つ(図 12)。センサから取得した情報を Android 端末で集約する。使用している OS のバージョンは Android 5.0.2 である。

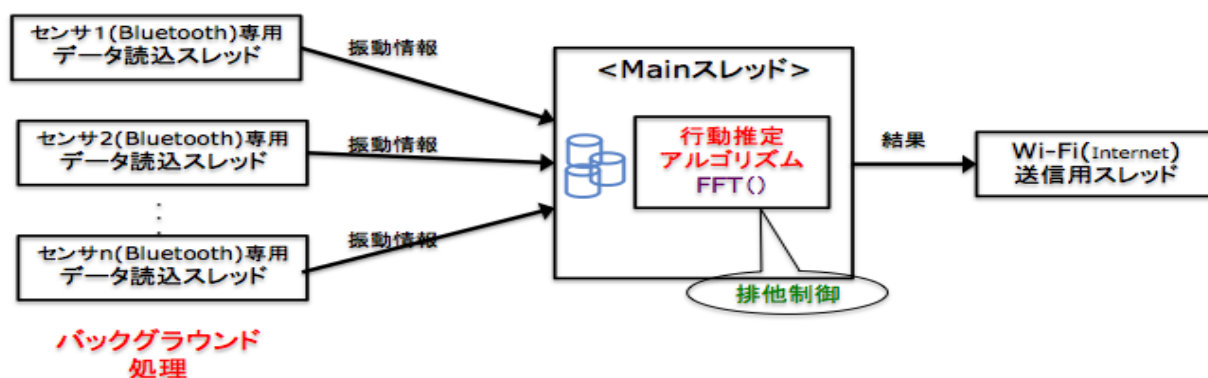


図 12 Android アプリケーション内の構成

Android 端末での受信処理は図 13 に記載されているフローチャートのように行われる。あらかじめ、振動データを受信するには Android 端末上から対象のセンサとペアリング動作を行う。受信処理は Android アプリケーションとして実装する。

Android アプリケーションで実行されるプログラムの処理は以下のようになる。

1. センサデータを順次受信（読み込み）する
2. 垂直方向（Z 軸）のデータのみ抽出する
3. バッファに 1 分間分のデータが溜まったかを確認する
4. 1 分間データが溜まったら行動解析を行うためのフーリエ変換を行う
5. 1～3 or 1～4 の処理を繰り返し行う

行動解析が終わったらデータは順次サーバなどに解析結果のデータを送信する。

Bluetooth モジュール (RN-42) は Bluetooth SPP というシリアル通信でデータを送信するため、Android アプリケーションでは Bluetooth 通信機能の UUID を SPP のプロファイル

「00001101-0000-1000-8000-00805F9B34FB」に設定しなければならない。UUID を正しく設定しないと正常に通信することが出来ない。

Google Play Store[12]で既存の Bluetooth SPP 対応のアプリケーションを複数ダウンロードし Android 端末で加速度センサから振動データの受信を試みた。しかし途中の振動データが欠落する事象が発生した。原因は振動センサのサンプリング周期が速いため、Bluetooth の中間バッファがあふれたためであった。

次に Android のサンプルプログラム (Android SDK) で Bluetooth 通信の代表的なサンプルプログラムである BluetoothChat を使用して受信実験を試みた。BluetoothChat の場合、データを取りこぼしてしまい正しい振動データを受信することが出来なかった。

そこで、加速度センサから振動データを取りこぼさずにデータの受信が行えるためソースコードの改良と取りこぼしの原因を解明しながら実装を行う必要がある。

図 13 は実装した Android アプリケーションのシーケンス図である。

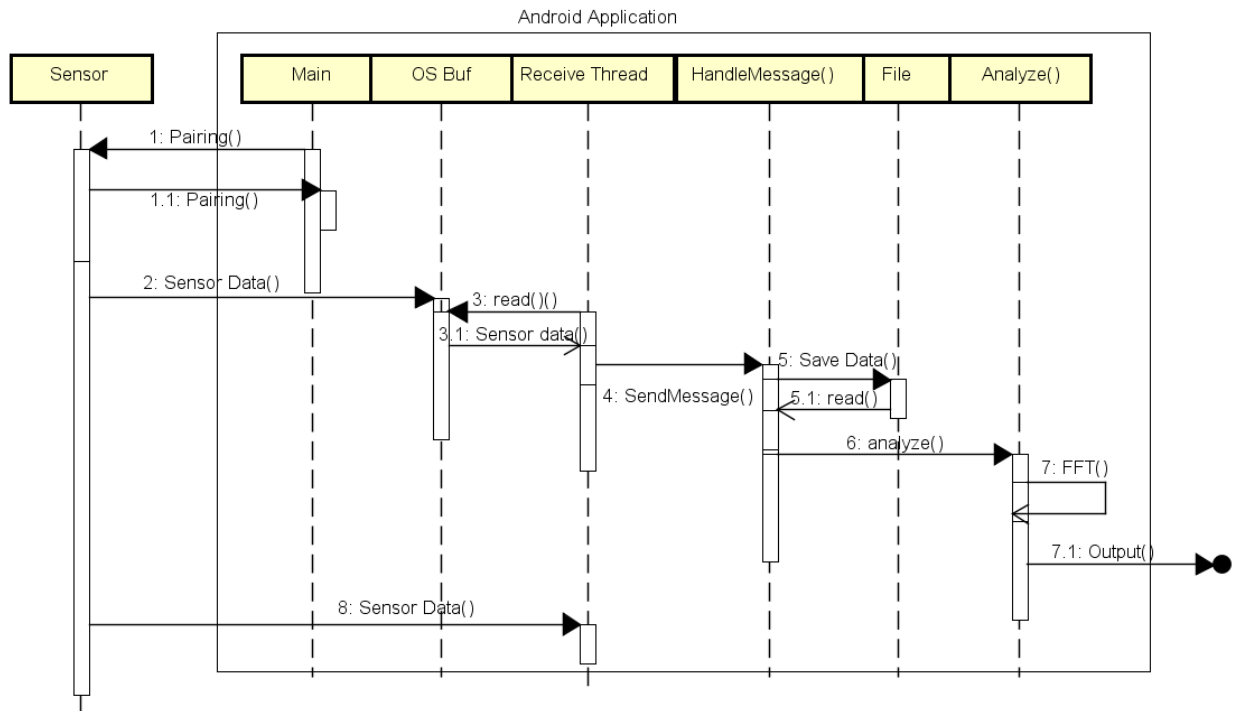


図 13 Android アプリケーションのシーケンス

シーケンス図の具体的な処理は、

1. BluetoothSocket を作成し振動センサに接続要求を行う
2. センサから Android OS の中間バッファにデータが送信される
3. 中間バッファから Android アプリケーション上のバッファにデータを読み込む(read0)
4. 読み込んだデータをオブジェクト化し Handler へと渡す
5. Handler で 1 分間のデータを貯めるまで 1～4 を繰り返す
6. 1 分間のデータから垂直方向 (Z 軸) だけ正規表現で取り出す
7. 行動解析を行い (振幅の最大値確認・フーリエ変換) サーバに解析結果を送信する

3.6.3.Android アプリケーション部の処理（受信処理）

Android アプリケーションの動作の肝となる受信処理と実装上の工夫について説明する。Android アプリケーションは既存のアプリケーション BluetoothChat を元に実装を行った。しかし、このソースコードにはいくつか解決しなければいけない問題があった。問題が引き起こす取りこぼし・不具合の原因は主に 3 つある。受信バッファのサイズ、受信バッファからの読み込み動作を行うスレッド待機時間、Byte 型から String 型にキャスト変換時のソースコード上の不具合である。

3.6.3.1.受信バッファのサイズ・読み込みスレッド待機時間の問題

最初に受信バッファサイズの問題について言及する。振動センサから送る 1 サンプルのデータを計測した結果は 34 バイトであった。振動センサはサンプリング周波数が 100[Hz]のため、中間バッファに送られてきたデータを読みこむ時間が 1 秒（1000 ミリ秒）の場合（1 秒おきに読み込む時）、100 サンプルのデータが送られるので $34 \times 100 \text{byte} = 3400 [\text{bytes}]$ になる。中間バッファ（ストリーム）からデータを受信する場合は、Android アプリケーションのバッファサイズを最低でも 3400[bytes]以上の大きさにしなければならない。もし、バッファのサイズが小さい場合、常に中間バッファにデータが溜まり続けてバッファオーバーフローが起きてしまう（図 14）。また、スレッドの待機時間が長いと中間バッファ（カーネル空間のバッファ）のデータを正しく読み込まない時があるため、適切な時間を設定する必要がある。

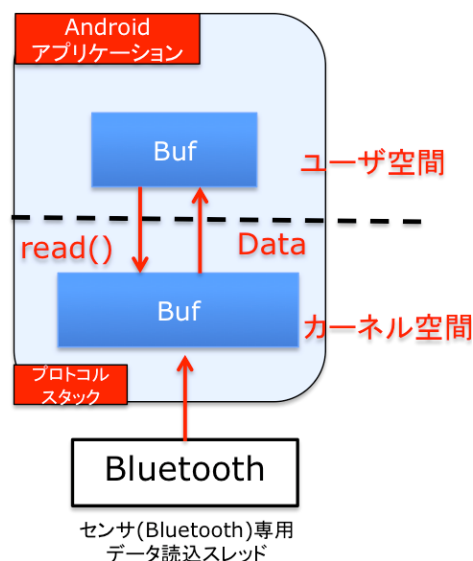


図 14 中間バッファから振動データを読み込む処理

3.6.3.2.Byte 型から String 型へのキャスト変換問題

OS の中間バッファから Android アプリケーションのバッファを読み込む時、中間バッファにデータが無い時は読み込み動作を行わず、中間バッファにデータが格納されている時は格納されているデータ分または Android アプリケーション上のバッファの大きさ分読み込み動作を行う。読み込み動作を行う場合、Android アプリケーション上で用意しておくバッファサイズは余裕を持った大きさにしておく。Android SDK のサンプルプログラムである BluetoothChat では仮に 1024[Byte]のバッファで 600[bytes]の読み込み動作を行った時、残りのメモリ領域に未定義データが残ったままの状態では Byte 型から String 型にキャスト変換を行っている。そこで受信したバイト数分のバッファを新しく定義して再代入することで上記の問題を解決した（図 15・図 16）。

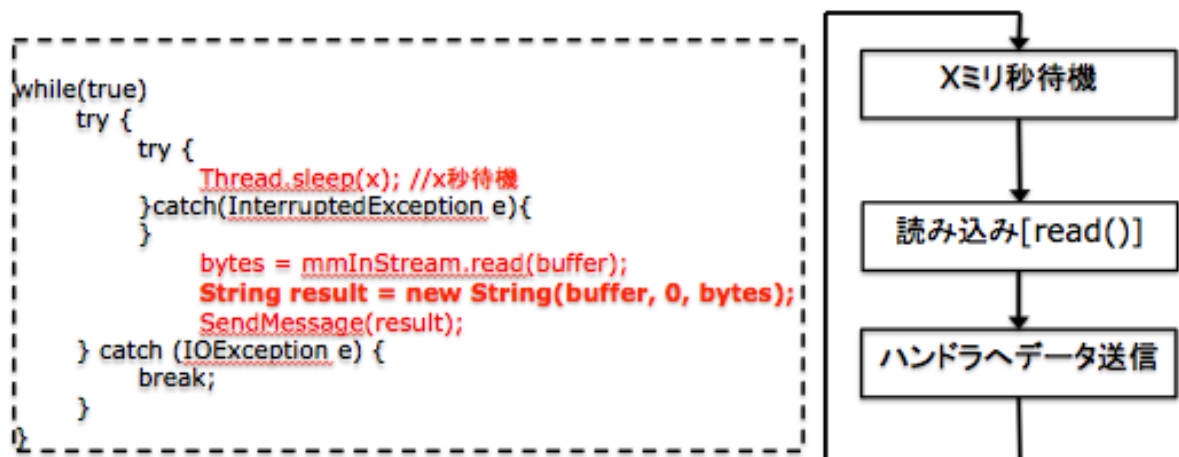


図 15 読み込み動作

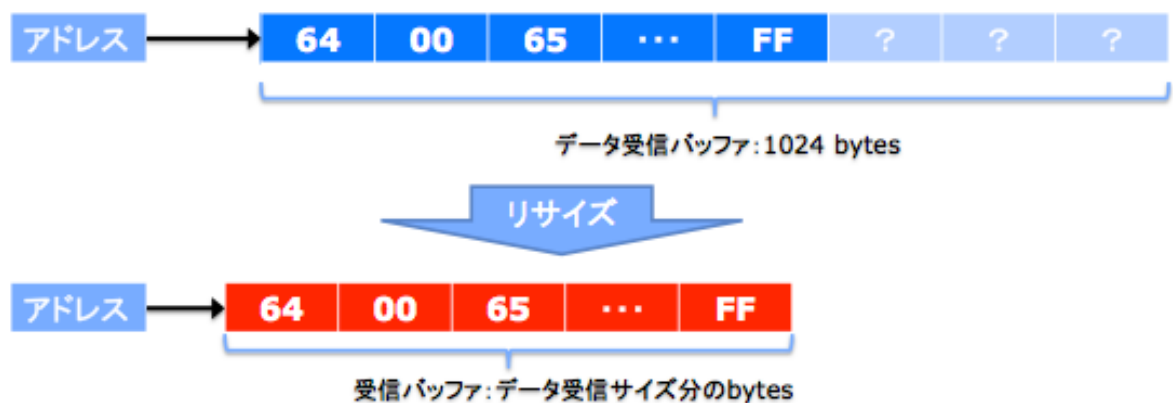


図 16 バッファサイズの調整

3.6.4. Android アプリケーション部の処理（複数センサから同時受信）

複数の振動センサから同時受信を行う場合、Bluetooth のピコネットという機能を用いる。ピコネットの最大接続台数はマスターが 1 台、子のスレーブが 7 台である（図 18）。本研究では Android 端末をマスター、振動センサをスレーブとして動作させることで Android 端末を使って複数センサからの受信処理を行う。マスターである Android 端末ではそれぞれのスレーブ（センサ）に対し受信スレッドを作成する。それぞれのセンサを区別するためにはセンサ ID を使用する（図 19）。それぞれの受信スレッドがメッセージ・キューにデータを送信する。ハンドラは随時到着したメッセージから随時、センサ ID と一致したファイルにデータを格納していく。これより、センサ毎に独立したデータをファイルに格納可能することで、センサ毎に行動解析が可能である（図 20）。

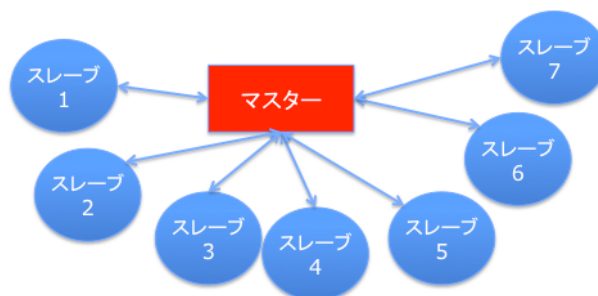


図 17 Bluetooth のマスター・スレーブ関係（ピコネット）

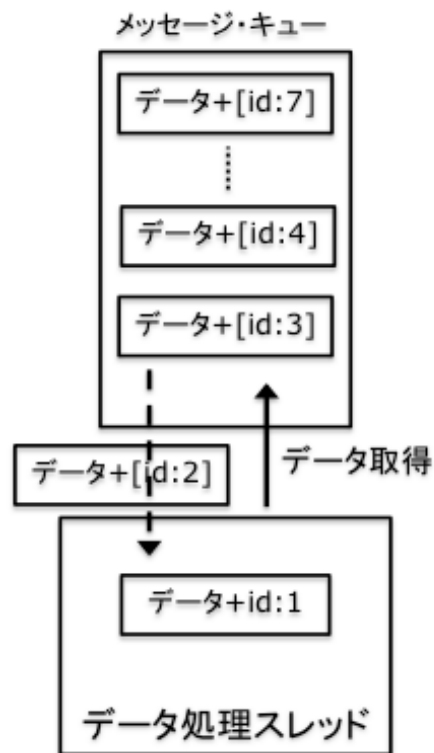


図 18 センサ ID を使った受信処理

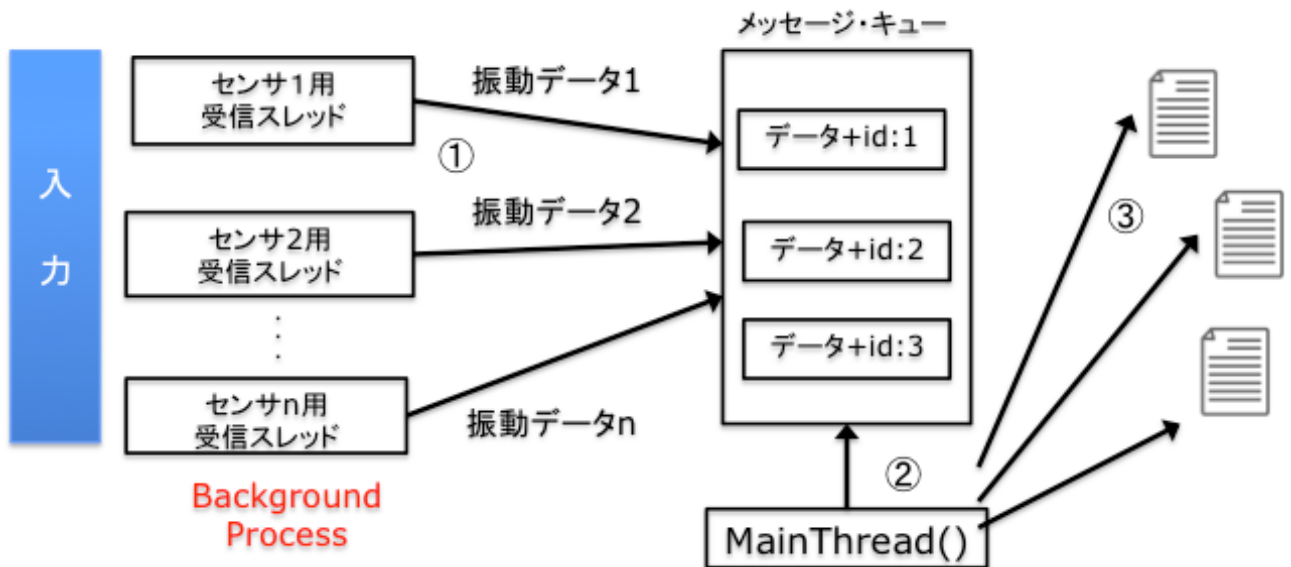


図 19 複数の受信スレッドとメッセージ処理受信

3.6.4.1. 受信処理（データ処理スレッド）

データ処理スレッド内のハンドラはメッセージ・キューからデータが順次渡される。センサデータが 60 秒分蓄積したら行動判定を行わなければならない。しかし、行動判定の計算処理は時間がかかる。行動判定中は、データ処理用バッファ・データ蓄積用ファイルは使用されているため、センサ 1 台に対しそれぞれバッファとファイルを 2 つずつ用意し、行動解析中・データ受信用に分けてデータを格納することでデータの不整合を回避した。(図 21)

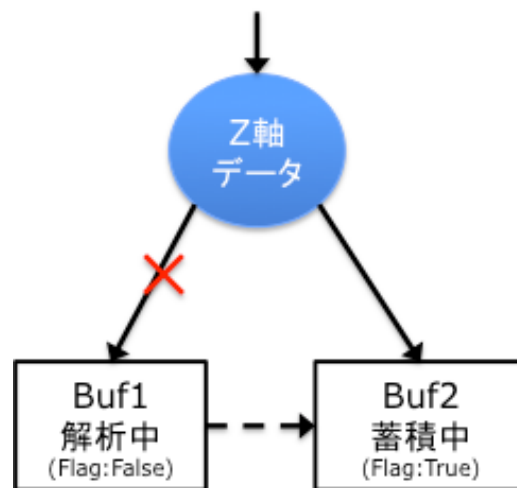


図 20 ダブルバッファリング

4. 評価実験

4.1. 実験の目的

高齢者に負担の少ない見守りシステムを作るためのプラットフォームを作成するにあたり、複数の振動センサからデータを正しく受信でき、振動データから歩行者の行動情報の特徴を振幅から確認することを目的とする。

4.2. 実験 1 : 1 台のセンサと Android 端末の Bluetooth 通信実験

部屋の床に 1 台の振動センサを設置し、Android 端末で正常にデータの受信が出来たかの確認を行う。送信データは Raspberry Pi で制御している加速度センサの値を画面に出力した結果をみる。Raspberry Pi に HDMI ケーブルを差し込み画面で確認し、受信データは Android 端末で受信処理を行った後に作成されたファイルを参照した。

送信データと受信データが合致していたため、受信処理後に作成された振動データが格納されているファイルをグラフに表示した。

実験は 1 K のアパートで行い、人間が行動をしない安静時と歩行時の 2 種類のデータを確認した。(図 22、図 23)

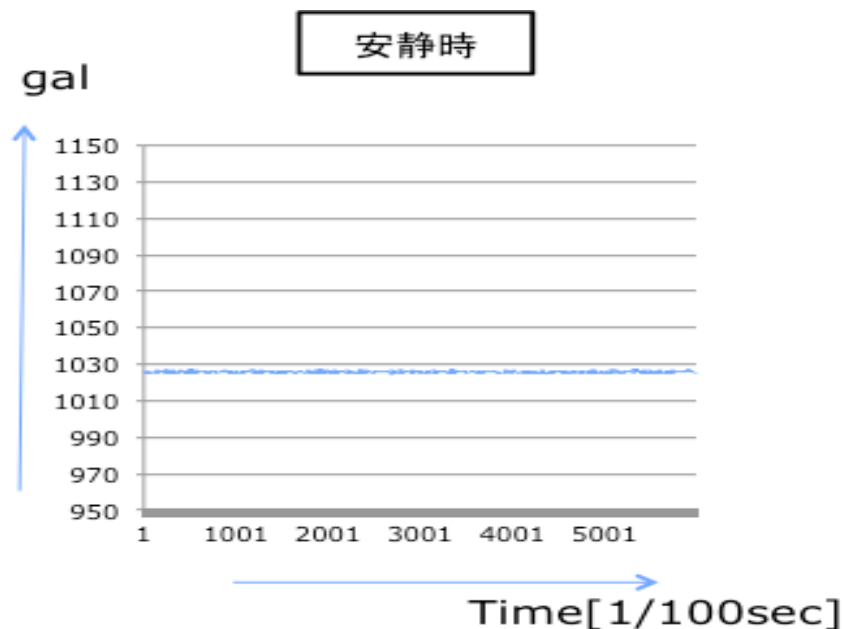


図 21 安静時の振動データ(センサ 1 台)

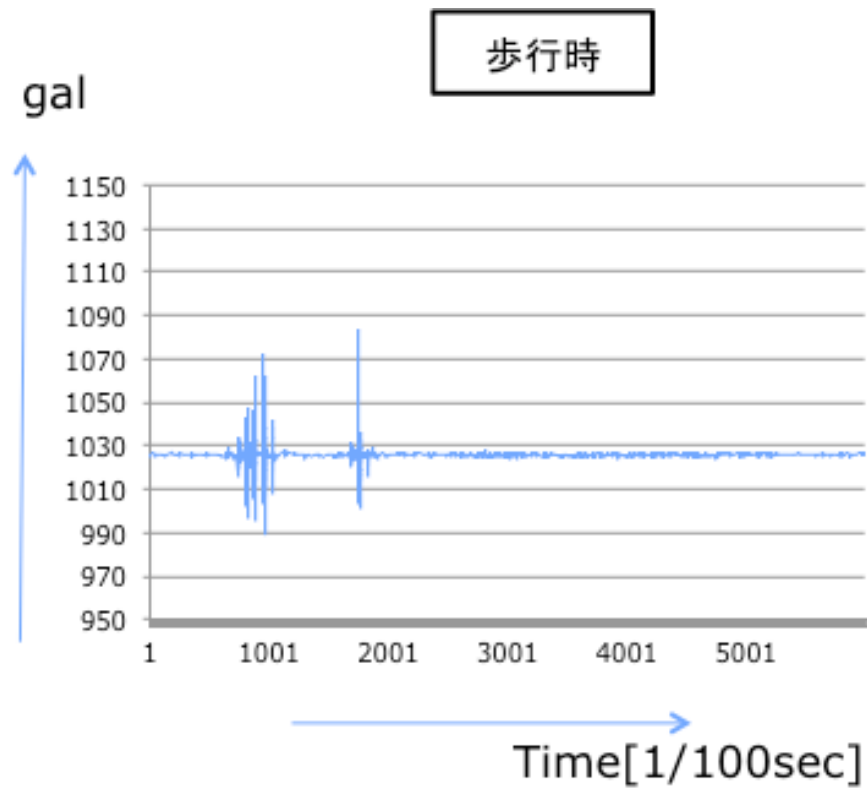


図 22 歩行時の振動データ(センサ 1 台)

次に図 22 と図 23 の波形をフーリエ変換した時に振動データにどのような特徴が出るかを確認した。フーリエ変換は本研究のために自作したモジュールで行った(図 24、図 25)。

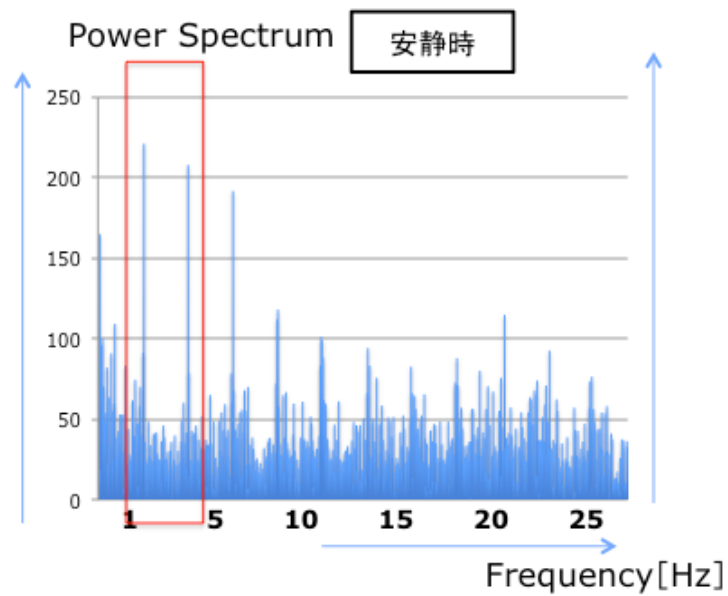


図 23 安静時の振動データの周波数解析(センサ 1 台)

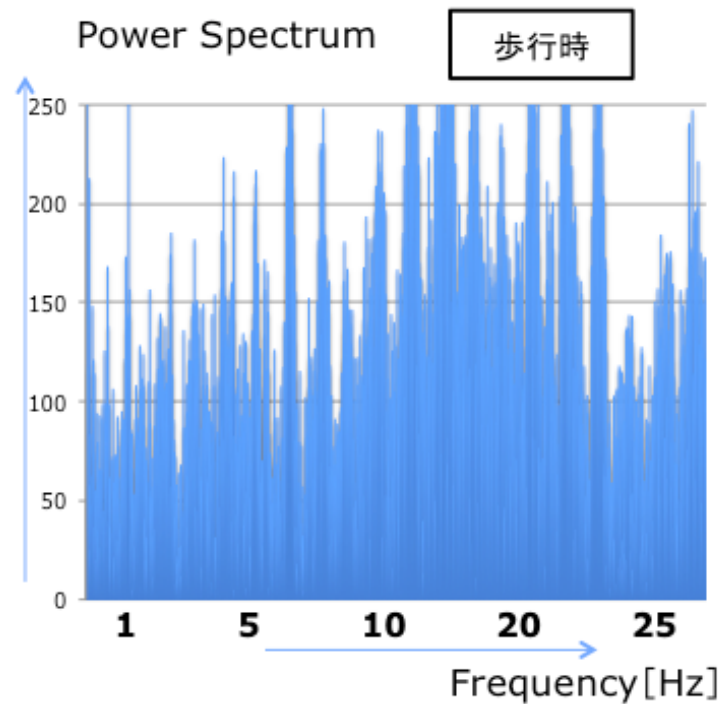


図 24 歩行時の振動データの周波数解析(センサ 1 台)

4.2.1.実験 1 の考察

図 22 と図 23 の結果、安静時には波形データに揺らぎがなくほぼ平行線のような状態であった。一方、歩行時の振動データは 10 秒と 20 秒の場所に振動のスパイクが見えた。これはある地点を人間が往復した時の振動波形である。これより、振動データから人間が歩いたことをしっかりと感知することが可能であることが判明した。

図 24 と図 25 の実験では、安静時・歩行時の振動データをフーリエ変換して周波数解析をした。安静時は約 3[Hz]の部分で通常の 4 倍近いパワースペクトルが見られる。これは建物で持っている特定の周波数で揺れやすいという固有周波数によるものである。地面の揺れと建物の固有周波数がちょうど一致するのが約 3[Hz]である。人間の行動だけでなく、建物自体の周波数成分の検出も出来た。歩行時の周波数解析のグラフでは、幅広い周波数帯でパワースペクトル成分が出た。歩行時とは明らかに違う成分のパワースペクトルが出力されていることが分かる。

4.3. 実験 2：複数台のセンサと Android 端末の Bluetooth 通信実験

部屋の床に複数台の振動センサを設置し、Android 端末で正常にデータの受信が出来たかの確認を行う。それぞれのセンサはほぼ同じような場所（近く）に置き、それぞれのセンサが同じような波形を示すかを確認する。振動センサは同時に振動を観測し、同時に振動データを Bluetooth 通信で送信している。実験は 1 K のアパートで行い歩行時の振動データを確認した（図 25）。

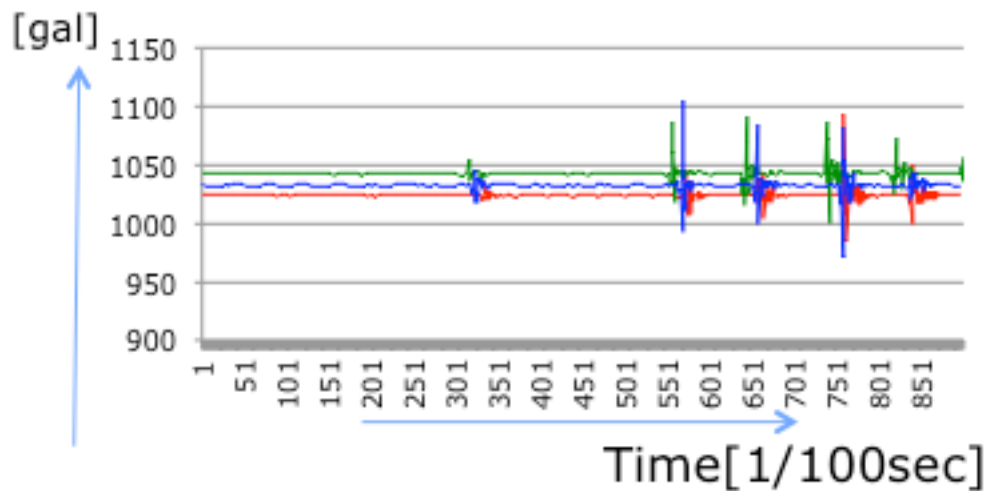


図 25 歩行時の振動データの周波数解析(センサ複数台)

4.3.1. 実験 2 の考察

3 種類の振動データを観測してみたところ、ほぼ同じ振動波形が表示された。これはセンサをほぼ同じ場所に設置したので正常な結果である。複数台のセンサからの Bluetooth 通信を Android 端末で正常に受信処理することが確認できた。

センサが持つ特性によって微妙に振動データの違いが出てくるが行動解析に関わるほど大きな変動ではないため特別な対策は要らない。

4.4. 実験 3：排他制御の実装による遅延の計測

Android 端末で振動データを常に取り組むための実装では、ダブルバッファリングを用いている。一つのバッファはユーザ空間からデータを読み込むバッファ、もう一つは振動情報から行動を推定するバッファとして用いている。2つのバッファは1分ごとに切り替わるため、1分以内に行動推定のような負荷の高い処理を終えなければならない。実装では行動推定をするバッファの処理に擬似的な重い処理として、フーリエ変換とフーリエ変換の結果をグラフとして Android 端末の画面に出力する（以下 UI 処理）プログラムを実装した。

センサ 1 台から振動情報を受信している時(1 スレッド)、フーリエ変換と UI 処理を行った時の処理にかかる時間を計測した。この処理はバッファが切り替わる 1 分以内に処理を終えなければならない。実験機器は SONY Xperia Z3 compact (Android5.0.2)を使用した。

計測の結果は 7.116 秒で処理を終えたため、1 分以内の処理を終えることを確認できた。

行動推定の処理には排他制御を行っているため、1 度に 1 スレッドのみしか行動推定処理が出来ないため、他スレッドには処理待ちの時間が発生する。

これらを、考慮すると行動判定の処理に合わせて、他スレッドが行動判定処理が終わるまで待つ時間を足した時間を 1 分で終えなければならない。

通信速度の都合上、センサの接続台数は 4 台のため、
行動判定の処理時間 7 秒 + 他スレッドの処理時間(3×7 秒) = 28 秒 < バッファの切替時間(1 分)

実装による遅延が起きるものの受信処理に対する影響を与えないことが分かったため、正常に受信処理が行えることが分かった。

5. おわりに

本稿では、高齢者に負担をかけない人間の行動推定システムを実用的なレベルまで構築するために、リアルタイムで人間の行動推定を行えるシステムのプラットフォーム部分を実装したことについて報告した。

既存の振動センサを使った高齢者見守りシステムでは一時の行動判定しか出来なかったが、本研究ではセンサからの情報を従来の有線ケーブルから Bluetooth 通信に置き換え、振動データは Android 端末で一元的かつ自動で処理することが可能になったことを示した。

上記の目標を達成するために実装上の様々な工夫（ダブルバッファリング・スレッド待機時間の変更・バッファサイズ・排他制御）を実装した。

実験を通して、実装したことが正常に動作していることを確認し、歩行時と安静時の見分け方や建物固有の特徴を抽出することが出来たことを示した。

謝辞

本研究を行うあたり、ご指導を頂いた岩手県立大学ソフトウェア情報学部の澤本 潤教授、杉野 栄二講師、マクロネットワークス熊谷様に心より感謝致します。また、学部時代から研究についてのアドバイスを熱心にしていただいた京都産業大学コンピュータ理工学部の瀬川 典久准教授に感謝いたします。

最後に今まで多大な支援と協力をしていただき、暖かく見守ってくれた両親・家族に深く感謝いたします。

参考文献

- [1] 総務省統計局, “統計局ホームページ/平成 25 年/統計トピックス No. 72 統計からみた我が国の高齢者 (65 歳以上) - 「敬老の日」にちなんで - II 高齢者の人口”,
<http://www.stat.go.jp/data/topics/topi721.htm>
- [2] 品川佳満, 橋本勇人, “人間性へ配慮した高齢者見守りシステムの開発”, 川崎医療福祉学会誌, Vol.11, No.1, 2001, pp199-204, <http://iss.ndl.go.jp/books/R000000004-I5902533-00>
- [3] 花水木法律事務所, 2008 年 9 月 24 日,
<http://hanamizukilaw.cocolog-nifty.com/blog/2008/09/post-f4ed.html>
- [4] 内閣府, “(1)主な収入源は年金が過半数”,
http://www8.cao.go.jp/kourei/whitepaper/w-2013/zenbun/s1_3_1_01.html
- [5] IO-DATA, TS-WLCAM/V, <http://www.iodata.jp/product/lancam/lancam/ts-wlcamv/>
- [6] 内田 泰広, 澤本 潤, 杉野栄二, “加速度センサを活用した非装着型の人間の行動推定システム”, HIP2015-72, pp1-6, <http://www.ieice.org/ken/paper/20150928MbCW/>
- [7] 朱牟田 善治, 小田 義也, 澤本 潤, 厚井 裕司, “据え置き型振動センサを用いた人の行動同定”,
https://www.jstage.jst.go.jp/article/ieejieiss/135/12/135_1583/_article/-char/ja/
- [8] 上田建揮, 玉井森彦, 安本慶一, “スマートホームにおける複数のセンシングデータに基づいた生活行動データ抽出”, マルチメディア, 分散, 協調とモバイル(DICOMO2014)シンポジウム, pp1884-1891,
https://ipsj.ixsq.nii.ac.jp/ej/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=105140&item_no=1&page_id=13&block_id=8
- [9] 太田 茂, 安井 秀作, 堀内 健司, “スマートホンを用いる高齢者書体の生活状況見守りシステムの開発”, 関西福祉大学社会福祉学部研究紀要, 16 巻, 1 号, pp67-73, 2012,
https://kusw.repo.nii.ac.jp/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=224&item_no=1&page_id=24&block_id=42
- [10] 大野 雄基, 手嶋 一訓, 加藤 大智, 鈴木 秀和, 旭 健作, 山本 修身, 渡邊 晃, “TLIFES を利用した徘徊行動検出方式の提案と実装”, 研究報告グループウェアとネットワークサービス (GN), 12 号, pp1-8,
https://ipsj.ixsq.nii.ac.jp/ej/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=88021&item_no=1&page_id=13&block_id=8
- [11] RN-42N, <http://ww1.microchip.com/downloads/en/DeviceDoc/rn-42-ds-v2.32r.pdf>
- [12] Google Play, <https://play.google.com/store?hl=ja>