

# 修士論文

## 圧縮センシングを用いた 行動モニタリングのための 携帯端末省電力化手法

2013年2月6日

指導教員 川原 圭博 講師

東京大学 大学院情報理工学系研究科  
電子情報学専攻 48-116402

明村 大登

## ■ 内容梗概

---

本研究では，携帯端末を用いてセンシングした加速度データをサーバに連続して送信するヘルスマニタリングなどのアプリケーションを前提に，送信が必要なデータ量を，圧縮センシングを用いて，低負荷に削減することで，携帯端末の消費電力を削減する方法について提案した．

携帯端末においてセンシングアプリケーションが消費する電力は，データ送信などの通信による部分が多い．そうした通信コストを削減するためには送信が必要なデータ量を圧縮することが必要になるが，圧縮に計算コストを多くかけた場合には端末の CPU による消費電力が無視できなくなるといったジレンマがある．そこで，圧縮が軽負荷に行える圧縮センシングの技術を用いて通信コストを低減することで端末の消費電力を削減する手法を提案した．圧縮センシングは圧縮が簡単な復元処理には複雑な計算を必要とするが，携帯端末で圧縮と送信を行い，復元処理はデータの送信先であるリソースの豊富なサーバ等に依頼することで，携帯端末側の消費電力を節約することができる．ただし圧縮センシングを用いる際には，センサデータのスパース（疎）性に応じて復元誤差が生じることが知られている．そこで，Human Activity Sensing Consortium (HASC) から提供される大規模加速度データベース HASC2010corpus を用いて，復元に伴う誤差と，その誤差が実際の行動認識にどの程度影響を及ぼすかを評価した．さらに，実際の Android 端末上にセンシングアプリケーションを実装し，その消費電力を調べることで本提案の有効性について評価を行った．

# 目次

---

|       |   |    |
|-------|---|----|
| 第 1 章 | 序論  | 1  |
| 1.1   | 本研究の背景                                    | 2  |
| 1.1.1 | 携帯端末の進化                                   | 2  |
| 1.1.2 | 参加型センシングの概念                               | 2  |
| 1.1.3 | 加速度センサを用いた人間の行動認識                         | 3  |
| 1.1.4 | 消費電力の課題                                   | 4  |
| 1.2   | 本研究の目的                                    | 4  |
| 1.3   | 本論文の構成                                    | 5  |
| 第 2 章 | 携帯端末を用いた参加型センシング                          | 6  |
| 2.1   | はじめに                                      | 7  |
| 2.2   | 参加型センシングの概要                               | 7  |
| 2.2.1 | 登場の背景                                     | 7  |
| 2.2.2 | 定義  | 8  |
| 2.2.3 | アプリケーション                                  | 9  |
| 2.3   | 筆者の取り組み：ウェアラブル血圧センサを利用した高齢者のヘルスマonitoring | 10 |
| 2.3.1 | 背景  | 10 |
| 2.3.2 | カフレスなウェアラブル血圧センサ                          | 11 |
| 2.3.3 | 行動データの自動認識                                | 11 |
| 2.3.4 | 行動認識アプリケーションの仕様                           | 12 |
| 2.3.5 | 参加型センシングへの応用                              | 13 |
| 2.4   | 実用化に向けた課題と研究事例                            | 14 |
| 2.4.1 | プライバシーの問題                                 | 14 |
| 2.4.2 | 消費電力の問題                                   | 15 |
| 2.4.3 | 本研究の位置づけ                                  | 16 |
| 第 3 章 | 圧縮センシングを用いた消費電力削減手法                       | 18 |
| 3.1   | はじめに                                      | 19 |
| 3.2   | 信号のスパース表現                                 | 19 |
| 3.3   | 圧縮センシングの数理                                | 20 |
| 3.3.1 | 概要  | 20 |
| 3.3.2 | 圧縮アルゴリズム                                  | 20 |
| 3.3.3 | 復元アルゴリズム                                  | 20 |
| 3.3.4 | 観測行列と基底行列のコヒーレンシ                          | 21 |
| 3.3.5 | 制限等長性                                     | 22 |
| 3.3.6 | 復元誤差                                      | 22 |

|              |                             |           |
|--------------|-----------------------------|-----------|
| 3.4          | 行列の選択                       | 23        |
| 3.4.1        | 観測行列 - sparse random matrix | 23        |
| 3.4.2        | 基底行列 - K-SVD                | 24        |
| 3.4.3        | K-SVD のアルゴリズム               | 24        |
| 3.4.4        | K-SVD を用いることのメリットとデメリット     | 26        |
| 3.5          | 参加型加速度センシングへの適用             | 26        |
| <b>第 4 章</b> | <b>復元誤差と行動認識率の評価</b>        | <b>28</b> |
| 4.1          | はじめに                        | 29        |
| 4.2          | データセット                      | 29        |
| 4.2.1        | HASC2010corpus の利用          | 29        |
| 4.2.2        | データの前処理                     | 30        |
| 4.2.3        | K-SVD による基底行列の構築            | 30        |
| 4.2.4        | 圧縮・復元を行ったデータセット             | 31        |
| 4.3          | 復元誤差の評価                     | 31        |
| 4.3.1        | NRMSE による交差検定               | 31        |
| 4.3.2        | 復元の例                        | 31        |
| 4.3.3        | 結果                          | 32        |
| 4.4          | 行動認識率の評価                    | 32        |
| 4.4.1        | 加速度を用いた行動認識の研究例             | 33        |
| 4.4.2        | 本研究における行動の認識手法              | 33        |
| 4.4.3        | 結果                          | 34        |
| 4.4.4        | 特徴量の値の変化                    | 35        |
| <b>第 5 章</b> | <b>消費電力の評価</b>              | <b>39</b> |
| 5.1          | はじめに                        | 40        |
| 5.2          | 実験概要                        | 40        |
| 5.2.1        | 各処理の消費電力                    | 40        |
| 5.2.2        | 全体の消費電力                     | 41        |
| 5.3          | 実験準備                        | 41        |
| 5.3.1        | ハードウェア・セットアップ               | 41        |
| 5.3.2        | ソフトウェア・セットアップ               | 42        |
| 5.3.3        | 3G によるデータ通信後の IDLE 制御       | 43        |
| 5.4          | 各処理の消費電力測定結果                | 44        |
| 5.4.1        | 加速度センシング                    | 44        |
| 5.4.2        | データ圧縮                       | 45        |
| 5.4.3        | データ送信                       | 46        |
| 5.5          | 全体の消費電力測定結果                 | 47        |
| 5.5.1        | 異なる圧縮方式における比較               | 47        |
| 5.5.2        | 考察                          | 47        |
| 5.6          | 評価のまとめ                      | 49        |

|                      |    |
|----------------------|----|
| 第 6 章 結論             | 50 |
| 6.1 本研究の主たる貢献        | 51 |
| 6.2 今後の課題            | 51 |
| 6.2.1 観測・基底行列のサイズ最適化 | 51 |
| 6.2.2 行動ごとの圧縮最適化     | 52 |
| 6.2.3 実環境試験          | 52 |
| 参考文献                 | 54 |
| 付録                   | 60 |

# 目次

|     |  |    |
|-----|--|----|
| 1.1 | 携帯端末を用いたセンシングの規模 [1]   | 3  |
| 1.2 | 参加型センシングにおける携帯端末の役割  | 4  |
| 2.1 | 参加型センシングの構成 [11]   | 8  |
| 2.2 | 参加型センシングのアプリケーション例   | 10 |
| 2.3 | ウェアラブル血圧センサを用いた高齢者の収縮期血圧の短期変動測定結果と行動ラベル  | 12 |
| 2.4 | Android 上で行動認識を行うデモアプリケーションの動作例  | 13 |
| 3.1 | 歩行中の加速度ノルムとそのスパース表現  | 19 |
| 3.2 | L-1 ノルム最小化のイメージ ( $K=1, m=2, n=3$ )  | 21 |
| 3.3 | 上位 20%のエネルギー成分による歩行中の加速度   | 23 |
| 3.4 | 圧縮センシングを用いた参加型センシング  | 27 |
| 4.1 | HASC2010corpus に含まれる行動の加速度データの例  | 30 |
| 4.2 | 圧縮率 30%において復元した歩行信号のスパース表現   | 32 |
| 4.3 | 様々な圧縮率で復元した歩行信号  | 36 |
| 4.4 | { 静止, 歩行, 走行, スキップ, 階段昇降 } の 6 行動を K-SVD により構築した行列を用いて復元したときの平均誤差と圧縮率の関係         | 37 |
| 4.5 | 圧縮率と認識精度の関係  | 38 |
| 4.6 | 行動認識に用いた特徴量の値と圧縮率の関係   | 38 |
| 5.1 | Android 端末の消費電力計測に用いた機器 (docomo GALAXY NEXUS SC-04D と Agilent 34410A デジタルマルチメータ) | 43 |
| 5.2 | 3G によるデータ通信における通信デバイスの状態変化   | 44 |
| 5.3 | 3G によるデータ送信にかかる消費電力  | 45 |
| 5.4 | 加速度センシングの 4 つのサンプリング周波数とその消費電力   | 46 |
| 5.5 | 圧縮方式ごとのデータ圧縮におけるデータサイズと消費電力量の関係  | 46 |
| 5.6 | 3G によるデータ送信におけるデータサイズと消費電力量の関係   | 47 |
| 5.7 | ZIP 圧縮と圧縮センシングの圧縮とデータ送信にかかる消費電力の比較   | 48 |
| 5.8 | 圧縮方式の違いによる端末の消費電力の内訳の比較  | 49 |

# ■ 表 目 次

---

|   |    |
|---|----|
| 4.1 非圧縮の加速度データを用いて行動認識を行ったときの認識精度 (Overall: 76.9%)    | 35 |
| 5.1 docomo GALAXY NEXUS SC-04D の主要スペック [48] . . . . . | 42 |

# 第1章

---

序論

## 1.1 本研究の背景

### 1.1.1 携帯端末の進化

スマートフォンをはじめとする携帯端末には、加速度センサ、GPS、マイクといったセンサが内蔵されるようになった。近年では気圧センサや温度センサなど、内蔵されるセンサの種類も多様なものとなってきている。こうしたセンサは、元々は電話としての機能に必要であったり端末の操作性を改善したりするために内蔵されたものであったが、こうしたセンサを人間の行動やそれを取り巻く環境を知るために用いることが増加してきている。具体的には、端末を持つ人間のとっている行動、車の渋滞状況、気候の変化などである [1]。

それに加えて、携帯端末にはセンサだけでなく CPU やメモリ、Wi-Fi や携帯電話回線による無線機能も備わっている。また、市場に出回っている主要な携帯端末においては、SDK や API、開発ツールなどが既に用意されており、センシングアプリケーションの構築を容易に行える基盤が揃っている。Apple Appstore や Google Play などのようにアプリケーションを全世界へ配布する場も用意されているため、専用の無線センサノードに頼ることなしに、大規模なセンシングやデータの分析が以前よりもはるかに容易に行えるようになった。何よりも特徴的な点としては、携帯端末は文字通り人々が日々携帯し、持ち歩くものであるということである。世界中の人間が持ち歩く端末を用いてセンシングしたデータを共有することができれば、広範囲に渡るデータを利用して、場所や個人に応じた細やかなサービスを提供することができ、快適なユビキタス社会の実現に近づくと考えよう。

### 1.1.2 参加型センシングの概念

携帯端末のセンサを応用するアプリケーションにも、その規模によって様々なものがある (Fig. 1.1) [1]。センサ情報を元に携帯端末の持ち主の運動状況やカロリー消費量を測定、管理して健康を促進するアプリケーションも多く、こうしたものは個人的な用途のアプリケーションである [2]。

個人的な用途以外に、主に近接したいくつかのセンサ同士がグループとなって、協調した動きをすることで、センシングにかかる電力を抑えたりセンサデータの信頼性を高めたりする手法も存在する。例えば Miluzzo らは、それぞれのセンサがある程度信頼のおける判定モデルを持っていることを仮定した上で、そのモデルと個人の判定結果を共有し、それらを乗算して全体での判定を行うことで推定精度を高める Darwin と呼ばれるプラットフォームを実装している [3]。Darwin では、何か特別な興味を持った複数の人の持つ携帯端末が近い位置に居るときに、1 台のセンサによる推定結果が間違っている場合、他のセンサのモデルや推定結果を共有して用いることで全体としては高い精度の結果を得ることができる。このようないくつかのセンサによる協力によるセンシング方法は協調センシング (Cooperative Sensing、もしくは Collaborative Sensing) と呼ばれ、室内での話者認識などに応用されている。

更に範囲を広げて、近接しているかどうかに関わらずより多数の人間がデータを収集するコンセプトの代表例として、参加型センシング (Participatory Sensing) という概念がある [5, 6]。人々が何気なく携帯端末を持って日常生活を送るときに様々なセンサを駆動しておくことで、それが動くセンサとなって環境の情報や人間の活動そのものをセンシングすることができる。主に人口密度の高い都市部においてそうした情報を人々に定期的に提供してもらい、収集し、遠隔のサーバで解析を行うことでより快適な日常生活を送るために有益なサービスを提供しようというコンセプトである。UCLA の Center For Embedded Networked Sensing (CENS) を中心に様々なシナリオとアプ

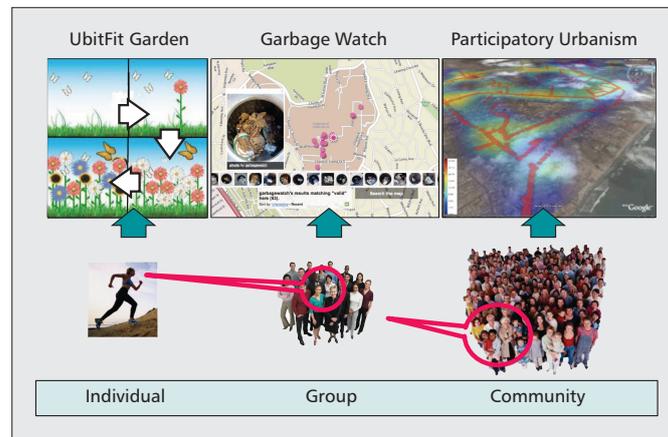


図 1.1: 携帯端末を用いたセンシングの規模 [1]

リケーションが考案されており [4]，天気，車の渋滞予測，ライフログなど，その分野は多岐にわたる．

こうした参加型センシングのアプリケーションは人々の携帯端末を用いて行われるため，固定型のセンサノードを用いる場合と比べて以下のようなメリットがある．

- 電池の取り替えを行う必要が無い．
- 広範囲のデータを扱うことができる．
- 人の流れや行動などを認識することができる．

本研究は，こうした大規模なセンシングアプリケーションを対象として行う．

### 1.1.3 加速度センサを用いた人間の行動認識

端末に内蔵された多くのセンサの中でも，加速度センサは最もよく用いられるセンサの一つである．元々，端末に内蔵された加速度センサは携帯端末のスクリーンの向きを判定してユーザインタフェースを向上するためのものであった．しかし，近年，加速度センサを端末の持ち主の行動認識に応用しようという試みがなされてきている．参加型センシングのアプリケーションの中には，気圧センサや温度センサなどを用いて周囲の環境を知るアプリケーションに加えて，こうした加速度センサ，GPS，ジャイロセンサといった動きを検知するセンサを用いて端末の持ち主の行動を認識するアプリケーションが多く存在している．行動認識の技術は参加型センシングと組み合わせることで，ライフログや健康促進のサービスに応用することができる．代表的な例として，Nike 社が提供するセンサ端末やアプリケーション，ならびに Web サイトなどの総合的な名称である “Nike+” がある [7]．端末の持ち主は加速度センサや GPS による個人の運動の記録をアップロードし，それらを共有することができる．自身の運動データを他の参加者と共有し，競争する，1 日の運動の目標を定めてそれを達成する，ジョギングに適切なルートを教え合うといった行動を行うソーシャルネットワーキングサイトとなっており，世界最大のランニングコミュニティとなっている．実用化されている参加型アプリケーションの一つと言える．

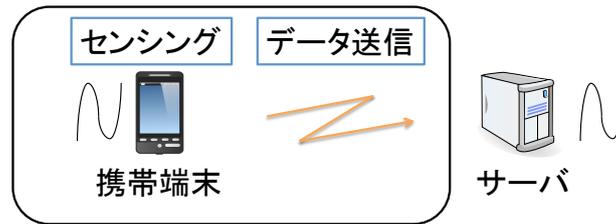


図 1.2: 参加型センシングにおける携帯端末の役割

#### 1.1.4 消費電力の課題

こうした参加型センシングによるサービスの質を高めるには、集まるデータの量を多くするために、より多くの参加者にセンサデータを提供してもらうことが必要となる。そのためには前述した行動認識のアプリケーションのように、ゲーム性を持たせる、ユーザ間の競争を促すなど、参加者にインセンティブを持たせるサービスを開発することが望ましい。そして、もしユーザにインセンティブを与えられなかった場合でも、センシングとそのデータ提供に必要なとされるコストはなるべく小さく抑えられるべきである。参加型センシングのアプリケーションは多く考案されているものの、実用化に至っているものがまだ多くない理由として、この参加によるコストが大きいことが挙げられる。参加コストには、センサデータという個人情報を提供することによるプライバシーの問題などが含まれ、そうした実用上の問題の中で最も大きな課題は、携帯端末の消費電力である。スマートフォンを始めとする携帯端末にはセンシング以外にも通話や Web ブラウジングといった主要なタスクが存在するため、バッテリーは逼迫している。ユーザが参加型センシングアプリケーションに参加するデメリットを感じないようにするためにも、こうした副次的なタスクにはなるべく電力がかからないことが望ましい。以上のような理由から、参加型センシングにおいては携帯端末側の消費電力を低減する方法が必要となっている。

## 1.2 本研究の目的

以上の議論を元に、本研究では、参加型センシングアプリケーションにおいて、センシングとデータの送信を行う携帯端末の消費電力を削減することを目的とする (図 1.2)。中でも特に、加速度センシングを行いそのデータの送信を行うアプリケーションに着目する。

後述するように、日常的な用途において携帯端末における消費電力は、端末内部の計算処理よりもデータの送受信による電力消費が大きな割合を占める。本研究で取り扱う参加型のアプリケーションのように連続的にセンシングとデータ送信を行う状況を想定した場合にも、データ送信にかかる消費電力は非常に大きなものとなる。データ送信のコストを下げるためには、送信が必要なデータのサイズ、すなわちセンサデータのサイズを減らすことが必要となる。そのための方法として、センサデータをなるべく生に近い状態で送信するためには、データを圧縮することが考えられる。しかしながら、複雑なデータ圧縮のアルゴリズムを携帯端末内に実装した場合、圧縮処理の計算に負荷がかかることによって CPU の消費電力が増加してしまうというジレンマがある。そのため圧縮処理にはなるべく電力を消費しないことが望ましい。

そこで本研究では、圧縮センシングの技術を使うことで低負荷にセンサデータの圧縮を行い、携帯端末の消費電力を低減することを提案する。圧縮センシングは、Candés らによって確立されたスパース (疎) 性を持つデータに対して適用可能な圧縮・復元技術であり、その復元の処理には複

雑な計算を必要とする反面、圧縮は非常に軽負荷に行えるという特徴がある。これを携帯端末に適用することで、受信側であるサーバでは複雑な処理が必要となるものの、圧縮にはコストをかけずに送信が必要なデータサイズを小さくできるため、携帯端末側の消費電力を低減できる。

### 1.3 本論文の構成

以上の提案に基づき、本論文の構成は以下のようになっている。

- 第1章 序論
- 第2章 携帯端末を用いた参加型センシング
- 第3章 圧縮センシングを用いた消費電力削減手法
- 第4章 復元誤差と行動認識率の評価
- 第5章 消費電力の評価
- 第6章 結論

第2章では、携帯端末を用いた参加型センシングについてその概要を述べる。これまでに提案されているアプリケーションの例に加えて、本研究において取り組んだ高齢者のヘルスマニタリングについて説明する。そして参加型センシングの実用化に必要な問題として消費電力の問題を取り上げ、消費電力低減のためのこれまでの研究について内容をまとめる。

第3章では、携帯端末における参加型センシングの消費電力低減のための方法として、圧縮センシングを用いることを提案する。まず、圧縮センシングの数理についてその圧縮・復元アルゴリズムの数理をまとめる。圧縮センシングを信号に適用する場合には、圧縮に用いる観測行列と復元に用いる基底行列などを定める必要があるが、その2つについてそれぞれ説明する。特に基底行列については、基底行列の構築手法の1つであるK-SVDと、大規模加速度データベースHASC corpusを用いて、人間の行動の加速度データをスパース(疎)に表現することに適した基底行列を作り出す方法を提案している。この詳細について説明し、この圧縮センシング手法を用いることでどのように携帯端末が省電力化されるかについても触れる。

第4章では、提案した手法の評価を行う。圧縮センシングではその原理上、信号のスパース性によって復元誤差が生じることが知られているが、その誤差が圧縮率によってどのように変化するかを評価した。また、実際に行動認識を行うようなアプリケーションを想定した場合に、圧縮・復元を行った信号を用いて行動認識手法を適用した場合、圧縮が認識率にどのように影響するかを評価した。ある程度まで圧縮率を下げても、行動認識率を維持することが可能であることを示す。

第5章では、提案手法を用いた場合に携帯端末がどの程度消費電力を低減できるか、実際のAndroid端末上で行った評価について説明する。また、その評価実験の前準備として行ったこととして、携帯端末の通信デバイスのアイドル時間制御や消費電力の測定を行うためのハードウェアなどについて説明する。

第6章では、本研究の提案について結論づけ、今後の課題について触れる。

## 第2章

---

# 携帯端末を用いた参加型センシング

## 2.1 はじめに

スマートフォンをはじめとする携帯端末には多くのセンサが内蔵されるようになり、その端末としての用途はもはや電話や Web ブラウジングだけにとどまるものでは無くなりつつある。2012年には7億台を超える携帯端末が出荷されており、一人一台のスマートフォンを持つ時代が来たと言って過言ではない。持ち歩かれる多くの携帯端末からセンサ情報をボランティア的に受け取り、データを収集し、分析を行うことができれば、場所、時間、個人に最適化されたアプリケーションを考案することでユビキタス社会に貢献することができる。本章ではこうした考えから生まれた「参加型センシング」について概要を説明する。そのアプリケーションについていくつか例を挙げ、筆者自身のアプリケーション考案への取り組みについても紹介する。その実用化に向けて残された課題についてプライバシーの問題や消費電力の問題を取り上げ、それぞれの先行研究についてまとめる。

## 2.2 参加型センシングの概要

### 2.2.1 登場の背景

人間の活動とその周囲の環境情報を容易にセンシングできるようになった要因の1つとして、携帯端末の性能の進化が挙げられる [1]。スマートフォンを代表とする近年の携帯端末には、GPS、加速度センサ、カメラ、マイクロホン、光センサ、近接センサなど多数のセンサが内蔵されている。それに加えて Wi-Fi や携帯電話回線による無線通信機能が備わっており、専用の無線センサノードに頼らずとも普段持ち歩く端末を用いて容易に人間の活動や環境情報が観測できるようになった。

携帯端末のハードウェア自体の高性能化に加えて、主にスマートフォンでは SDK や API、開発ツールが用意されているため個人が容易にアプリケーションを開発できる。また Apple Appstore や Google Play などのように、開発したアプリケーションを全世界へ提供する場が用意されている。これらのおかげで、端末所有者がアプリケーションをダウンロードしてインストールするだけで様々なセンシングシステムが利用可能となり、大規模な実験も以前より遥かに容易なものとなった。

携帯端末の進化に加えて、Twitter や Facebook をはじめとするソーシャルサービスが近年流行し、企業もそれと連動したメディアの製作を精力的に行っている。ソーシャルサービスはまさに人間の活動の記録を残すユーザ生成型のデータベースであり、多くの人間が自らの興味に関するテキストや場所情報などを投稿することによって多数の人間とインタラクションを得る文化が流行している。そうした文化が、他者とのデータの共有を行うアプリケーションが注目され始めた1つの要因であると言える。

そして、そうしたソーシャルサービスは携帯端末におけるセンシングと相互に連携し始めている。Eagle らや Cranshaw らは被験者達の実世界の行動情報と交友関係を元に、Facebook などのウェブ上のソーシャルネットワークでの交友関係を推測する手法を提案している [9] [10]。Cranshaw らは被験者達が存在した頻度に応じて地図上の場所にエントロピーを定義し、エントロピーの低い場所で同時刻に存在した被験者同士は友人関係にある確率が高い、エントロピーの高い場所に居る頻度が高い人間は Facebook 上で友人の数が多く、といった統計的な結果を示しており、こういった結果をソーシャルネットワーク上での友人の推薦機能にフィードバックするといった利用も考えられている。

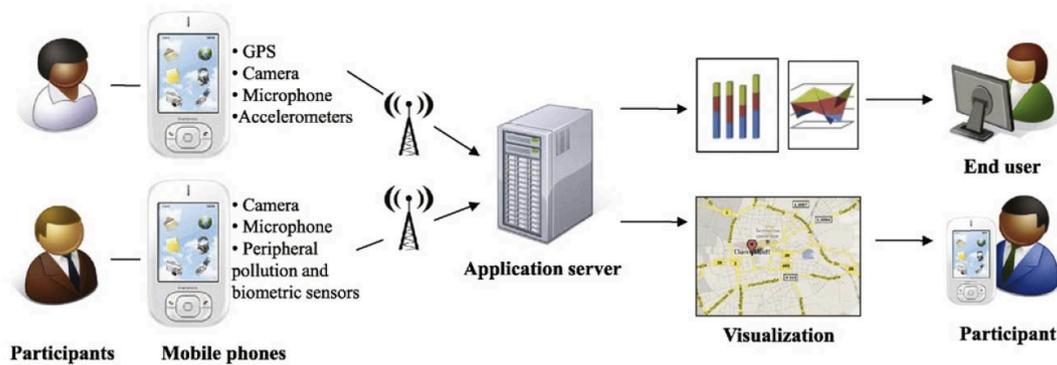


図 2.1: 参加型センシングの構成 [11]

ソーシャルネットワーク上に投稿されるテキストや画像もさることながら、連続して多数の人間から収集される携帯端末からのセンサデータの情報は今後爆発的に増加すると考えられる。以前はプライバシーの観点から、また端末にかかる通信コストを少なくするために他のマシンにセンサデータの解析を委譲することはしていなかったが、近年はサーバにそういった処理とサービスの提供を依頼するアプローチが主流となってきている [1]。家庭向けのインターネットの通信の進化に伴って、一般ユーザがネットワークを経由して高性能なアプリケーションを動作させる方法としてクラウドコンピューティングが注目されている。クラウド側にセンサデータを送信することで、センシングアプリケーションはリソースの少ない携帯端末側にかかるデータ処理などの計算の負担を減らすことが可能である。多くの携帯端末からセンサデータを収集し、クラウドで処理を行った結果に基づいてユーザにフィードバックを行うことで、よりパーソナライズされたサービスが提供可能となる [11]。こういった状況が、携帯端末を用いた大規模なセンシングが近年盛んに研究されている背景にある。

### 2.2.2 定義

こうした背景を元に、Burke, Estrin らは、参加型センシング (Participatory Sensing) の概念を提唱した [5, 6]。高機能化した携帯端末を移動するセンサノードとして用いて、携帯端末の持ち主からボランティア的にローカルなセンサデータを遠隔のクラウドに収集する。そうして草の根的に集められたデータはクラウド上で分析され、新しい発見と共にセンシング参加者へと提供される。こうしたプロセスのことを参加型センシングと呼ぶとしている (図 2.1)。どのような場所のどのようなセンサデータを募集するといった 1 つの要望をキャンペーンと呼び、Estrin は参加型センシングのモデルケースとなるキャンペーンは以下の 4 つの役割によって実現されるとしている。

1. (Initiators) センシングキャンペーンを開始し、キャンペーンの仕様を決定する。
2. (Gatherers) 携帯端末の所持者であり、キャンペーンに参加して、ユーザの意思やネットワークの使用状況に基づいてセンサデータを提供する。
3. (Evaluators) データを検証し、種類ごとに分類する。
4. (Analysts) データを解釈し、結果を報告し、結論づける。

ただし、キャンペーンはこれだけの役割で完成するわけではない。どのようなプロトコルでデータの収集を行うかといった課題以外にも、センサデータを収集・提供する方法、またそれをサーバサイドで分析する方法など、アプリケーションレベルの開発が当然必要となることに触れている。

### 2.2.3 アプリケーション

参加型センシングを提唱する事を動機づけたアプリケーションとして、Estrin は以下の4つの例を挙げている [6]。

1. Public health : 端末の持ち主から集められた健康状態に関するデータや行動データが多く集まれば、その統計データを分析することで病気の徴候を発見できる。また参加者にフィードバックを行うことで、病気の予防や自己の健康管理に使うことができる。
2. Urban planning : 既存の地理情報システム (Geographic Information System, GIS) と参加型センシングが連携することで、環境雑音の大きさなどを地図上に表示することができる。そうしたデータが都市の再計画へと役立てられる。
3. Cultural identity and creative expression : 地方において写真や音声をネットワークを通じて残すことで、そのデータ自体が地方の歴史やアイデンティティを示す文化的に重要なものとなる。
4. Natural resource management : 環境学者や生態学者の中では、フィールドワークによるデータの記録をいまだに紙の上を書くことで行っている事が少なくない。携帯端末のカメラ、マイク、GPS 等を使えば、取得できるデータの量だけでなくその質も向上できることが予測され、複数人でのデータの共有もしやすくなる。

その他にも、UCLA の Center For Embedded Networked Sensing (CENS) を中心に多くのセンシングシナリオとアプリケーションが考案されており [4]、重複するものもあるが、種類として人間の活動をセンシングするものと周囲の環境の状態をセンシングするものに大別できる。参加型センシングにおいて考案されているアプリケーションをその二つに分けて紹介する。

#### 人間行動に関するアプリケーション

まずユーザのヘルスケアに関するアプリケーションが挙げられる。Reddy らは、参加者達の食事の画像やそれに対するコメントなどによって参加者の健康を促すプロトタイプシステムを構築している [12]。あらかじめ被験者の体調を把握している状態から、被験者の行動や他の被験者と接触したイベントなどのデータを元に特別な健康診断を行う事無く個人の健康状態を予測する実験を行うなど [13]、様々な健康促進のためのアプリケーションが提案されている。

ソーシャルサービスとの関連では、Dartmouth 大学の CenceMe Project [14] において、被験者が他の人と会話中であるかどうか、座っているのか歩いているのか、などのセンサデータや計測値から推定される情報を Facebook や MySpace などの既存の SNS と共有するシステムの研究開発が行われた (図 2.2(a))。Gaonkar らは Micro-blog と呼ばれる、携帯端末からの場所や WiFi の SSID、加速度信号などの情報を加えた従来よりバラエティに富んだブログサービスを開発している [15]。

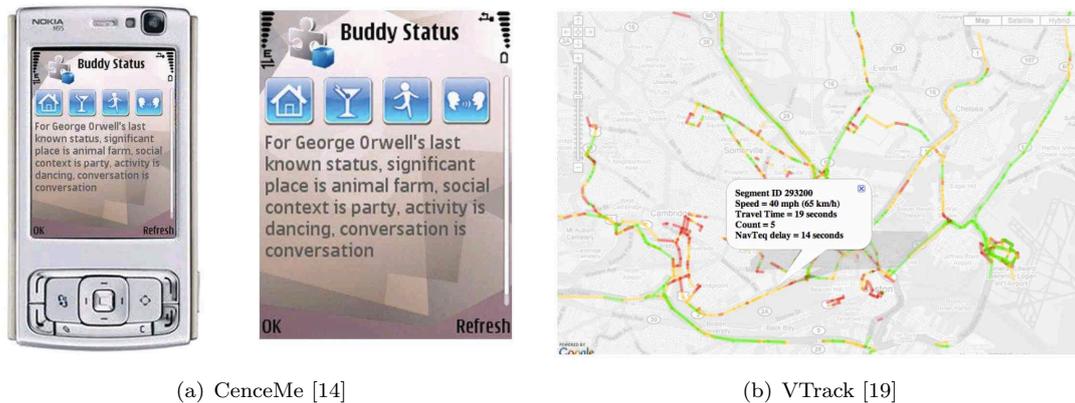


図 2.2: 参加型センシングのアプリケーション例

その他では, Deng らは参加者から投稿される画像を元に食料品や日用品の特売を発見するシステムを作り実験を行うなど, 人間の社会的な活動のセンシングに基づいたアプリケーションが多く登場している [16].

#### 環境情報に関するアプリケーション

人間の活動以外に, 公共における環境の情報をセンシングすることでより生活に有益な情報を提供する手法がある. Eisenman らは自転車で移動する人の場所に加えて加速度信号などを計測し, 個人の状態だけでなく路面状況など環境の状態を推測して自転車通勤者にとって最適なルートを導くシステムを提案している [17].

また, 都市部における環境音を観測して有用な情報を抽出するアプリケーションも多い. 騒音問題ではその状況と程度の把握のためには様々な時間帯における環境音を街全体で集めることが必要になるが, 新規にセンサを街に設置してデータを集めることはコストが高い. Kanjo [18] は参加者から街中の環境音を収集し, 騒音をモニタリングして既存の地図にその情報をマッピングするシステムを提案している.

携帯端末の位置は所有者に密着しているものばかりではない. 車の中にある携帯端末が GPS を用いて位置の変化から交通の渋滞状況を推測し, 目的地までの移動にかかる時間を推定するという用途も提案されている (図 2.2(b)) [19].

## 2.3 筆者の取り組み: ウェアラブル血圧センサを利用した高齢者のヘルスマニタリング

以上で紹介したアプリケーションに加えて, 筆者は独自に参加型のアプリケーションについて検討を行った. その概要について説明する.

### 2.3.1 背景

未曾有の高齢化が進む中で, 高齢者の慢性疾患管理が問題となっている [20]. なかでも高血圧罹患率は非常に高く, 高血圧を基盤とする様々な疾病を予防するには普段からの厳格な管理が必要と

なる．血圧の自動調節が破綻し過度の血圧変動が起きれば立ちくらみやめまい，ひいては転倒リスクにまでつながる．これらの現象は高齢者本人の生活の幅や質を大きく損ねるきっかけにもなり得るため，なるべく高齢者の血圧変動を連続的に，かつ負担をかけずに，臨床診療における高齢者の健康管理に活用する方法が求められている．また，高齢者においては就寝時，起立時，食事後などに劇的な血圧変化が起き，こうした短期的な血圧変動が上記のような症状を引き起こしやすくなる症例がある．症状を予防するためには，個々の患者の短期的な血圧変動のモニタリングを行うと同時にその行動情報を検出し，関係性を探ることが重要である．

### 2.3.2 カフレスなウェアラブル血圧センサ

従来の血圧測定においては，カフと呼ばれる袋状のベルトを圧迫することによって測定を行っていた．しかし，従来のカフ式血圧測定では頻繁な測定にも限界があり，同時にカフ圧迫という患者への負担も増える．また実際の臨床の現場においては，医療機関において診察にきた患者に対して1回の測定を行うだけであるため，個々の血圧変動の状態を把握することができていない．それに加えて，カフ式血圧測定が始まると被験者は行動を中断して安静にしなければならないという測定上の制限も出てくるため，行動と関連した血圧変動を測定することが難しい．

そこで，東京大学大学院工学系研究科・山田一郎研究室はカフを必要とせずに脈波伝播速度を用いて連続的に血圧をモニタできる血圧計を開発し，臨床への応用を進めている [21]．この原理は脈波伝播速度法を採用している [22]．脈波伝播速度法では，心電の R 波と脈波の立ち上がり点の時間差である脈波伝達時間 (Pulse Arrival Time: PAT) から，収縮期血圧値を算出することで血圧推定を行っている．脈波の計測部位として，体動による変化を最低限に抑えられる目的で耳たぶを選択している．上記の基本計測原理の検証のため，エルゴメータを用いた自転車こぎ運動による評価実験を行い，医師によるカフ式手動血圧計での聴診法による測定結果と比較して，大きな乖離のない結果が得られている．

### 2.3.3 行動データの自動認識

上述したウェアラブル血圧センサを用いて，実際に実験協力者の高齢者の収縮期血圧の短期変動を測定した結果を図 2.3 に示す．黒い点で表示されているのは，カフを用いて測定したときの血圧値である．実験中はウェアラブル血圧センサを身につけて様々な行動をとってもらう．各時刻において，とっていた行動は手動で記録しておいてもらう．図 2.3 の上部に，実験協力者の血圧測定後の自己申告によってつけられた行動ラベルを示している．歩行や階段昇降などの運動を行ったとき，休憩状態と比べて血圧が上昇していることが分かる．

こうした血圧の超短期変動は従来のカフ式血圧測定では測定できるものではなかったため，こうしたデータはまず大量に集めてみた後に分析を行い，何らかの知見とともに医師や実験協力者にフィードバックがなされることが必要である．最終的には，血圧データと行動データから緊急性の高い情報を読み取り，患者の健康に関わる問題を未然に防ぐ事が望ましい．ただ，血圧データはそれだけではなく，そのときどういった行動をとっていたかという情報とともに分析を行わなければ，意味の少ないデータとなってしまう．運動をすると血圧が高くなるといったことはこれまでも知られていたことであり，身体に異常が存在することを意味するとは限らない情報であるためである．また，行動ラベルを手動で記入する方法は手間がかかる事に加え，患者の自己申告に依存するため正確性を欠いたデータとなる事が懸念される．そのため，これらの血圧データと行動データは，携帯端末に含まれるセンサなどを活用することで，同時に，自動的に計測・分類されるべきで

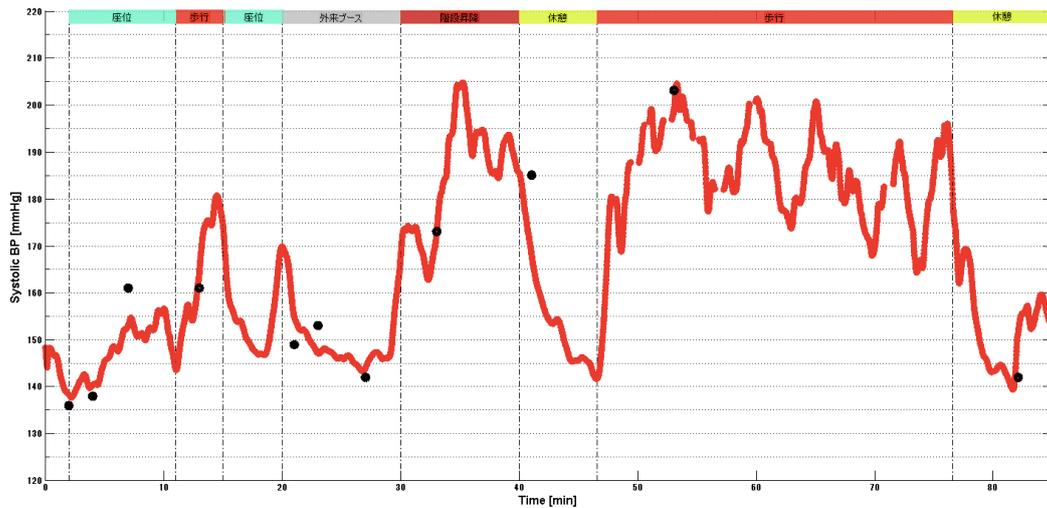


図 2.3: ウェアラブル血圧センサを用いた高齢者の収縮期血圧の短期変動測定結果と行動ラベル

ある．そこで，筆者は酒造正樹先生，Guillaume Lopez 先生らと協力し，携帯端末を用いて行動認識を行うアプリケーションを実際の Android 端末内に実装した．

### 2.3.4 行動認識アプリケーションの仕様

Android 端末上に，加速度センシングとその加速度データからの行動認識を行うデモアプリケーションを実装した．実装やテストには本稿の第 4 章，第 5 章のシミュレーションにも用いた docomo GALAXY NEXUS SC-04D を用いており，開発は Eclipse 3.6 (Helios) 上で Android SDK 4.0 を用いて Java でコーディングを行っている．デモアプリケーションでは以下のような順序で処理が行われる．

1. センシングをスタートするボタンがタップされると，その 5 秒後から約 124Hz で加速度センシングを行い，3 軸加速度データの  $l^2$  ノルムを計算した後リストに保存していく．
2. 7 秒に 1 回，リストから 7 秒分のセンサデータを取得し，窓幅 3 秒，スライド幅 1 秒で矩形窓を用いて 5 つのウィンドウを切り取る．
3. 5 つのウィンドウに対して特徴量を抽出し，分類器にかけて { 静止，歩行，走行，階段昇降 } の 5 行動の中から分類した結果を得る．
4. 得られた 5 つの結果から，多数決によって最も多く分類された行動のうちの一つをその時刻においてとっていた行動と結論づけ，画面に行動を表示する．
5. センシングを終了するボタンがタップされると，分類した行動の全てをログの形で表示する．

切り取る窓幅は 3 秒分，約 124Hz のデータであるが，必ずしも  $3 \text{ 秒} \times 124\text{Hz} = 372$  次元のデータとはならない事に注意しなければならない．後述するように，Android においては加速度センシングのサンプリング周波数を数値で指定することは出来ず，端末内で行う処理の負荷が高い場合に周波数が落ちることがあるからである．本アプリケーションにおいては，センシングをなるべく妨げ



図 2.4: Android 上で行動認識を行うデモアプリケーションの動作例

ることが無いよう、特徴量抽出と行動認識を行うスレッドをセンシングと別にしてある。また、認識の見かけの精度を高めるため、5つの窓から多数決を用いているようにしている。

行動認識には、機械学習用の総合ツールキットである Weka を Android に移植したものである Weka for Android ライブラリを用いた [23, 24]。訓練データには、筆者が同じ端末を用いて事前に測定した各行動 5 分ずつのデータを用いている。そのときの測定は、ズボンの左腰ポケットに携帯端末を入れた状態で行った。特徴量や分類器は、第 4 章で行った行動認識率の評価に用いたものと同じであるため、そちらを参照されたい。

図 2.4 に、実装したデモアプリケーションの動作例のスクリーンショットを示す。端末内でセンシングしたデータは、7 秒に 1 度行動認識がなされて画面にフィードバックがなされる他、各時刻にどのような行動をとっていたかをセンシング終了後にログの形で参照することができる。デモアプリケーションであるため画面に分かりやすい行動の認識結果を表示しているが、実用する際には消費電力の観点からこうした画面はオフにされているべきである。

第 4 章で述べる評価と同じく、歩行と階段昇降を加速度ノルムのみから識別することは非常に困難である。また、年齢や性別、身長、体重によっても運動の個人差は非常に大きいため、分類器は個人に応じて最適化される必要がある。そのために、様々な人から集められる大規模データを訓練データに用いることが実用上望ましい。

### 2.3.5 参加型センシングへの応用

今回は実装の簡便さのため Android 端末内で行動認識を行うデモアプリケーションを実装したが、その認識精度はまだ不十分なものである。その要因としては、訓練データの不足が考えられる。分類器が端末の持ち主によって訓練され、個人に最適化されていくべきであるし、用いる特徴量も個人に応じて改良されていくべきである。また、より生データに近い形で血圧データとともに分析を行う必要があることから、こうしたデータは端末内で認識を行ってしまうよりも、一度遠隔のサーバなどに蓄積した上で分析することが望ましい。その後、医師や専門家にデータベースを提供することで何らかの知見が得られることも考えられる。

そこで、こうしたアプリケーションを参加型センシングへと応用することが考えられる。患者から得られた血圧データや加速度データの生データを分析することで、分類の精度を高めて患者個人や周囲の状況に特化した健康管理アプリケーションを提供することが可能になるとともに、これまで知られていなかった有用な知見がマイニングされる可能性がある。

## 2.4 実用化に向けた課題と研究事例

これまで、参加型センシングアプリケーションの例を筆者の取り組みも含めて紹介してきた。こうしたアプリケーションが実用化されるようになるには、アプリケーションに共通する課題としてプライバシーの問題と消費電力の問題を考慮する必要がある。それぞれについて説明する。

### 2.4.1 プライバシーの問題

大人数からデータを収集する参加型センシングにおいては、場所情報、音声、写真といった個人情報共有が多いため、常にデータを投稿するユーザのプライバシーが問題となる。参加型センシングはより多くのユーザに参加してもらうことでより大量のデータが得られることが有益であるため、プライバシーが侵害される可能性のためにユーザ数が伸びないとアプリケーションとして成功しない。音声や写真、時間、場所情報といった典型的な個人情報はもちろんのこと、最近では加速度センサのデータからも人間が歩いている、走っているといった基本的な行動を認識する技術も存在しており、一見個人情報とは関係が無いように見えても全てのセンサデータに注意を払う必要がある。

連続してセンシングを行いデータを送信している環境では、誰かからのデータのストリームが止まったという事実自体が個人が特定される要因となるし、個人が特定されなくても、データの値の変化は知られる（誰の体重かが分からなくとも、誰かの体重が上がっていることは分かる、など）ことも考えられる。加えて、センサを身につけているユーザの近くに居る人間が他人のセンサによって状況が読み取られてしまう（写真、音声など）ことで個人情報を流出してしまうといった受動的な損害の可能性が挙げられる。

Christinらは参加型センシングにおけるプライバシーを「どのセンサデータをどの範囲の人間に公開するかをユーザが全て決められることを保証する」とことと定義し、プライバシー侵害の脅威に対する対策方法をまとめた [11]。携帯端末側での取り組みと、データを受信するサーバ側での取り組みに分けて説明する。

#### 携帯端末側でのデータの処理

よく用いられるのは信号にランダムなノイズを加えることでデータの perturbation を行う方法であるが [25]、ノイズを含んだデータが多数集まると復元は容易に行われてしまう。信号のうち個人情報を含む部分を除去する（環境音の中から人の音声混じった部分を除去するなど）などの方法で匿名化する取り組みが各アプリケーションでなされており、またセンサデータの粒度を下げて（どの建物に居るかでは無くどの街に居るか、生の加速度データではなく動いているかいないのみ）からデータを送信することでデータサイズが小さくなり、通信に必要なコストが下がるとともに個人情報が流出することが少なくなる。

類似した考え方として、参加型のアプリケーションにおけるセンシングタスクを近接した複数の端末同士で共有し、グループ全体としての結果をサーバに返すことで個人のプライバシーを守る方

法が提案されている [26]。しかし常に近接した端末が近くにありグループを形成できるとは限らず、個人からのデータでも完全に匿名化できる機構が望ましい。上述した方法はしばしば信号の推定精度とトレードオフの関係にあるが、アプリケーションの用途によっては有効である。

#### サーバ側でのデータ分析

携帯端末で処理がなされていたとしても、サーバ側でサービスを提供するまでにデータの分析を行う必要がある。例えば個人の家などプライベートな場所の情報はそれ自体が個人を特定してしまう可能性があり、そういった情報を既存の地図情報と照らし合わせてサーバ側で隠す、消去するといった方法が必要になる。プライベートな場所の情報のみを自動的に隠す完全な機構は未だ出ていない。

ユーザが誰に情報を公開するかを、データ送信先であるサーバに手動で設定するという方法は容易であるが、参加型センシングのアプリケーションはデータ提供者である参加者が直接利益を受けるものではなくより公共の目的で用いられることも多く、ほとんどのユーザがデータの提供を拒む可能性がある。参加型センシングによるアプリケーションがユーザに悪影響を及ぼさず有益な情報を収集するにはなるべくユーザの意志によらない、能動的で強力なプライバシー保護を行う方法が必要である。

#### 2.4.2 消費電力の問題

参加型アプリケーションにおいて各アプリケーションは、より多くの参加者を集めるためにユーザインタフェースにゲーム性を出す [2]、他のユーザとの競争を促すなどの方法でユーザにアプリケーションに参加してもらうためのインセンティブを与えることが有効である。また、普段持ち歩く携帯端末には電話や Email、Web ブラウジングといったメインの作業が存在する。もし参加するユーザにインセンティブが与えられなかったとしても、こういったメインの作業を邪魔することが無く、ユーザがアプリケーションに参加することによるデメリットを感じることは無いようにアプリケーションを設計しなければならない。

以上のような理由から、参加型センシングを行うアプリケーションにおいてはプライバシーの問題に加えて、携帯端末にかかる消費電力コストをより少なくする手法が必要となっている。

#### センシングアプリケーションごとの取り組み

各アプリケーションの研究において、用途に応じて消費電力を下げる試みがなされている。特に GPS を用いた位置情報の取得は参加型センシングにおいて最も多く使われる、かつ非常に電力コストが高い操作であり、Wi-Fi スポットの位置情報や、消費電力の少ない加速度センサを用いた移動方向の推定を行うなどの補助的な方法で、精度をなるべく落とさずに消費電力を低減して位置を推定する手法が多く考案されている [27]。

また他のセンサにも共通する単純な方法としては、Duty Cycle を設けてセンシングを行う間隔を間引いてセンサを駆動する時間を少なくする方法が広く用いられている。消費電力が下がるとともに、観測するデータのサイズが小さくなってサーバに送信するコスト、リアルタイムなサービスを提供する場合の遅延が少なくなるが、信号の推定精度は落ちることになる。

### 参加型センシング全般における消費電力の削減

複数のアプリケーションに適用できる，一般的な参加型センシングにおいて消費電力を低減する方法としては，Musolesi ら [28] と Wang ら [29] の研究が挙げられる．

Musolesi らは携帯端末を用いて連続してセンシングを行いサーバにデータを送り続けるアプリケーション全般において，コンテキスト推定処理を携帯電話とサーバで切り分けることで低消費電力化を実現している [28]．

Wang らはアプリケーションに用いるセンサに応じて，考えられるユーザの状態を動き（歩いている，走っている，など），場所（家の中，道路上，など），環境の状態（騒がしい，静か，など）の 3 つの点で定義してリスト化した．そして状態に応じて必要なセンサだけを動作させるモデルを各アプリケーションに構築することで端末の消費電力を削減することに成功している [29]．

また，端末の消費電力はセンサの駆動時間だけでなく他のデバイスの駆動時間にも影響される．連続してセンシングを行うアプリケーションが今後増えてくることを想定して，Priyantha らは現在の携帯端末に内蔵される CPU などのデバイスが駆動し続けている時間によるオーバーヘッドが大きいことに着目し，頻繁に CPU をスリープさせたりセンサの Duty Cycle の管理を容易に行えるサポートを行うことで消費電力の削減を試みている [30]．

こうした方法は有効であるが，ユーザがある行動をとる時間の配分などの統計に基づいたり，行動の認識に必要なデータ量に基づいてセンサの周波数や駆動時間の最適化を行ったわけではない．Wang らは更に消費電力を最適化するために，ユーザの行動分析に基づいた機械学習などを適用することで，更にセンサの駆動時間を落とすことが必要だとしている [29]．

### データ圧縮による省電力化

主に無線センサネットワークの分野では，データ圧縮によって送信にかかるコストを下げ，消費電力を低減しようとする試みがなされてきた．しかしながら，その多くは無線センサノードが近接した環境に散在し，それらがマルチホップする環境を想定したものである．その代表的なものとしては，圧縮の段階において複数の無線センサノードのデータ間の相関を利用する Cristescu らの研究が挙げられる [31]．また圧縮センシングを利用したものでは，Luo らが大規模な無線センサネットワークにおいて初めて圧縮センシングの理論を適用したシステムを提案している [32]．しかし，こうした研究も地理的に近接したノード群による無線メッシュネットワーク上で定義されたものであり，携帯端末のように人々によって持ち歩かれ，プライベートな情報をやりとりするセンシングシステムには直接適用できるものではない．その他，Yang と Gerla は体に装着した複数の加速度センサつきのノードからシンクノードまでの通信コストを削減するために，圧縮センシングを用いることを提案している [33]．しかし，これはリソースの非常に限られた無線センサノードからの通信に焦点を当てたものであり，その消費電力の評価もソフトウェア上でのシミュレーションに留まっている．

### 2.4.3 本研究の位置づけ

これまで，センシングアプリケーションにおける端末の消費電力低減に関する先行研究についてまとめてきた．本研究はスマートフォンをはじめとする携帯端末 1 台においてセンシングした加速度データを，無線環境を通じて遠隔のサーバへと通信するときに，圧縮センシングを用いて加速度データを圧縮することで通信コストを低減し，端末の消費電力を削減しようとするものである．そ

のため、無線センサネットワークのように複数のセンサノード間での連携をとることはできない。また、加速度センサ自体は消費電力の高いセンサではないため、より消費電力の低いセンサを補助的に用いることは考えにくい。

ただ、センサの駆動時間やハードウェアについて条件を定めたものではなく、センシングとそのデータ送信に的を絞ったものであるため、上述したデューティサイクルの定め方や低消費電力なプロセッサの利用などと衝突するものではなく、同時に用いることができる方法である。また、他のデータ圧縮との併用も考えられる。圧縮センシングによって圧縮されたデータに対して既存の圧縮方法を重ねて用いることで、より圧縮率を下げる可能性がある。そうしたことを踏まえ、本研究は、人間の行動の加速度に焦点を絞り、行動の加速度データに最も適した形でデータを圧縮することで、端末の計算負荷を最小限に抑えながら消費電力を削減することを目的とした研究である。

## 第3章

---

# 圧縮センシングを用いた消費 電力削減手法

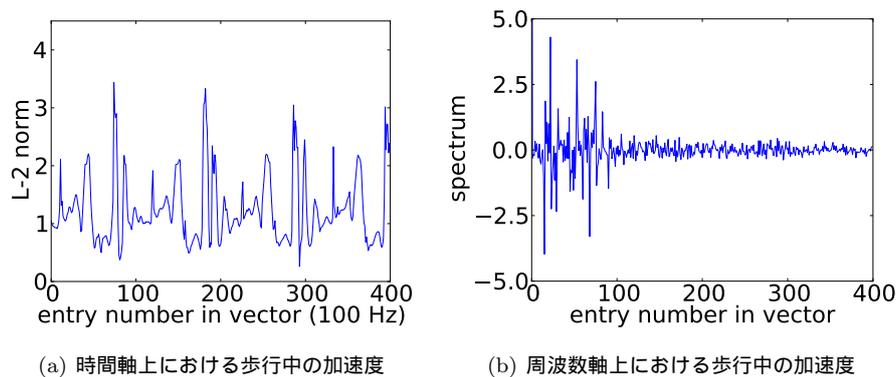


図 3.1: 歩行中の加速度ノルムとそのスパース表現

### 3.1 はじめに

本章では、提案の元となる圧縮センシングのアルゴリズムについて詳細に解説する。信号をスパースに表現することが注目を集めるようになった背景について紹介し、圧縮センシングの数学的な側面について説明する。圧縮センシングを適用する上で必要な観測行列と基底行列について、使用した行列とその構築方法についても説明する。特に、基底行列については、K-SVD と呼ばれるアルゴリズムを用いて人間の行動の加速度に適した基底の設計を行っている。最後に、圧縮センシングを携帯端末における加速度センシングに適用することでどのようなメリット・デメリットが得られるかを述べてまとめる。

### 3.2 信号のスパース表現

知覚において、多くのニューロンの中からあるパターンだけに特化した少数のニューロンのみが反応することで、多くのパターンを記憶したりパターンの重複を避けたりする戦略がとられている [34]。例えば、サルの視覚においては、「顔」に対してのみよく反応するニューロンが存在することが知られている。こうした情報の表現方法は「スパースコーディング」と呼ばれ、情報をスパース (疎) に表現することが注目されるきっかけとなった。

信号が何らかの基底上でほとんどの零成分とごく少数の非零成分で表現できるとき、その信号をスパースな信号と呼ぶ。信号のスパース表現は、情報保存量を増やす、複雑な情報の表現の簡略化、下流での情報の読み出しやすさ、エネルギーの節約のメリットがあり、信号処理の分野において注目されている。

例として、図 3.1(a) に、人間の歩行中における、ズボンの腰ポケットに挿入した携帯端末から得られる 3 軸加速度の  $l^2$  ノルムを示す。ある一定秒以上続く歩行信号は周期的な信号であることが分かる。この信号自体は時間軸上に散らばった成分を持っており、スパースな信号とは言えない。このデータに対して、離散コサイン変換 (Discrete Cosine Transform; DCT) を施すことで周波数軸上へ変換したときのスペクトルを図 3.1(b) に示す。ある一部の周波数上に成分が偏っており、少数の大きな成分と、多数の零に近い成分で表されることが分かる。歩行信号の加速度が、DCT 基底上ではある程度スパースな信号として表現できることが分かる。次節で説明する圧縮センシングはこうしたスパース性を持つ信号に対してのみ適用できる信号の取得方法であり、非常に少ない計算量で圧縮を行えることを特徴としている。

### 3.3 圧縮センシングの数理

#### 3.3.1 概要

信号を取得する際、従来は信号の最大周波数の2倍であるナイキスト・レート以上の周波数が必要であるとされてきた。現在の標準的なアナログデジタル変換 (ADC) も全てこのシャノンの定理に従っている。圧縮センシングはこの枠組みに反するデータ取得・圧縮方法であり、より少数の観測数から元の信号を復元できるとした理論である [8]。その条件として、以下の2つが挙げられている。

1. 元の信号が何らかの基底  $\Psi$  上でスパースな信号として表現できること。
2.  $\Psi$  上の成分は、観測するときに用いられた空間とはインコヒーレント (非干渉) であること。時間軸上でのスパイク信号が周波数成分上では分散するように、ある空間上での信号が別の空間上での表現では分散することをインコヒーレントであると呼ぶ。

観測する側は信号についての前知識が一切必要無く、信号の復元の際にのみ、基底  $\Psi$  を用いて計算することで復元を行うことができるのも圧縮センシングのメリットの一つである。以下でその圧縮・復元アルゴリズムについて説明する。

#### 3.3.2 圧縮アルゴリズム

元の信号を  $x \in \mathbb{R}^n$  としたとき、圧縮センシングにおける圧縮 (観測) は、 $m \times n$  の観測行列  $\Phi$  を用いて以下のように行われる。

$$y = \Phi x \in \mathbb{R}^m \quad (3.1)$$

これは  $x$  に関する  $m$  個の連立1次方程式を解くことに他ならず、 $m > n$  であれば大抵の場合解けることになる。 $m$  個の観測ベクトルの中からうまく  $n$  個を抜き出し、 $\mathbb{R}^n$  の基底とすることができればよいからである。圧縮を行う場合は  $m < n$  であり、式 3.1 は不良設定問題となる。疑似逆行列を用いてユークリッドノルムを最小化する解を推定することはできるが、一意には解けない問題である [35]。圧縮センシングの問題設定は、 $m < n$  のとき、元信号  $x$  がスパースに表現することができるからあらかじめ知っている場合に、 $y$  から元信号をいかに復元できるかというものである。

#### 3.3.3 復元アルゴリズム

正規化された  $n \times n$  基底行列  $\Psi$  を用いて、元信号  $x$  はスパースな信号  $s \in \mathbb{R}^n$  と表現できると仮定する。

$$x = \Psi s \quad (3.2)$$

$s$  のうち零でない成分の個数がたかだか  $K$  個であるとき、 $s$  は  $K$ -スパースであるという。 $s$  のうちどの成分が零でないかを知っている場合には、 $K < m$  であれば推定は簡単であるし、 $K < m$  であれば  $s$  を推定することはできない。ここでは、 $s$  のうちどの成分が零でないかが分からない場合を取り扱う。

復元は、線形制約  $y = \Phi x = \Phi \Psi s$  を満たした上でスパースな  $s$  を推定する問題であるため、以下のように表される。

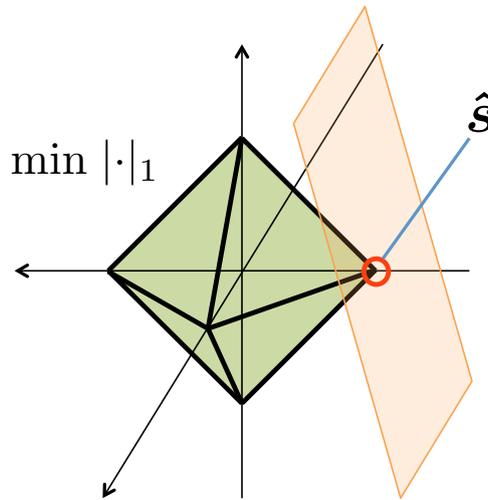


図 3.2: L-1 ノルム最小化のイメージ (K=1, m=2, n=3)

$$\arg \min_{\hat{s}} \|\hat{s}\|_0 \quad \text{subject to } \Phi\Psi\hat{s} = \mathbf{y} \quad (3.3)$$

$\|\cdot\|_0$  はベクトルの非零成分の個数を表す  $l^0$  ノルムであり、そのベクトルのスパース度合いを表す。膨大な  $s$  の非零成分の組み合わせが考えられるため、この  $l^0$  ノルム最小化問題は一般に離散最適化問題となる。 $n$  の値によって指数的に計算量が増加するため NP 困難であり、現実的には使用できない。そこで、以下の  $l^1$  ノルム最小化問題が用いられ、この解が  $l^0$  ノルム最小化と高い確率で一致することが知られている。

$$\arg \min_{\hat{s}} \|\hat{s}\|_1 \quad \text{subject to } \Phi\Psi\hat{s} = \mathbf{y} \quad (3.4)$$

$\|\cdot\|_1$  はベクトルの成分の絶対値の和を表す  $l^1$  ノルムである。この問題は  $l^1$  ノルム最適化、もしくは基底追跡法 (basis pursuit) として知られており、線形計画問題として最適化できるため、既存のシンプレックス法などを用いて効率的に解くことができる。本研究においては、MATLAB において凸計画問題のソルバである CVX を復元処理に用いた [36]。

こうして  $\hat{s}$  を推定し、既知の基底行列  $\Psi$  を用いて元信号  $x$  を復元することができる。K=1, m=2, n=3 における復元の例を視覚的に表すと、図 3.2 のようになる。図中で橙色の 2 次元平面が線形制約を表し、正しい解はこの平面中に存在する事を示す。緑色の正八面体は  $l^1$  ノルムが一定である条件を表す。この正八面体を徐々に大きくしていったときに初めて橙色の平面と交わる点、すなわち  $l^1$  ノルムを最小にする点が解となる。条件によってはこの  $l^1$  ノルム最適化が正しい解を与えない場合もあり得るが、後述するような適切な条件を満たす観測行列と基底行列が用いられているとき、ほぼ確実に  $l^1$  ノルム最適化が元のベクトルを推定することができる。

### 3.3.4 観測行列と基底行列のコヒーレンシ

観測行列  $\Phi$  と基底行列  $\Psi$  は、互いにインコヒーレントである必要がある。 $\Psi$  の第  $j$  列ベクトルを  $\psi_j$ 、 $\Phi$  の第  $i$  行ベクトルを  $\phi_i$  としたとき、 $\Phi$  と  $\Psi$  の間のコヒーレンシは以下のように定義さ

れる。

$$\mu(\Phi, \Psi) = \sqrt{N} \cdot \max_{1 \leq i, j \leq N} |\langle \phi_i, \psi_j \rangle| \in [1, \sqrt{N}] \quad (3.5)$$

$\langle \cdot, \cdot \rangle$  は内積を表す。すなわち、 $\Phi$  と  $\Psi$  の間に相関の高い列があった場合に、コヒーレンシは大きくなる。証明は省略するが、圧縮センシングはコヒーレンシの低い  $\Phi$  と  $\Psi$  の対によって成り立つ。 $\Phi$  には、一様分布の乱数行列、ガウス分布の乱数行列、 $\pm 1$  の値をとる乱数行列などを用いることで、ほとんどどんな  $\Psi$  に対しても低いコヒーレンシを示すことが知られている。

その他、コヒーレンシの低い  $\Phi$  と  $\Psi$  の組み合わせとして Candès は

1. 単位行列のランダム列を抜き出した行列による  $\Phi$  とフーリエ基底による  $\Psi$
2. noislets [37] による  $\Phi$  と Haar ウェーブレット、もしくは Daubechies D4, D8 ウェーブレットによる  $\Psi$

を挙げている。

完全な復元に最低限必要な観測数は、正の定数  $c$  を用いて

$$M \geq c \cdot \mu^2(\Phi, \Psi) \cdot K \cdot \log N \quad (3.6)$$

で定義されており、コヒーレンシの低い行列を選択することが、より圧縮率を良くするために重要な要素となる。

### 3.3.5 制限等長性

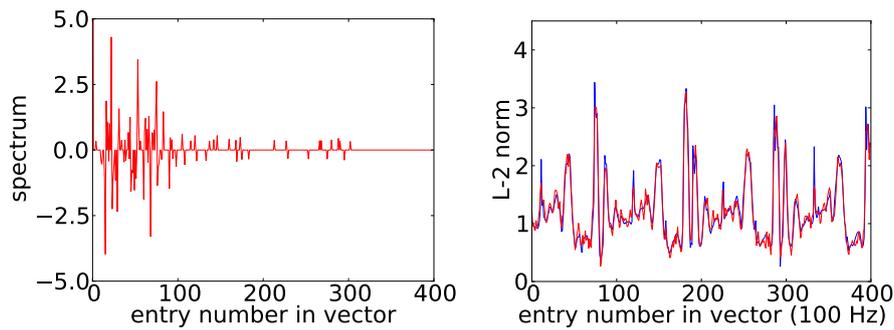
式 3.4 で示した  $l^1$  ノルム最適化が正しい推定結果を与える観測行列の条件について説明する。ある  $m \times n$  の観測行列  $A$  が与えられたときに、ある  $K$ -スパースなベクトル  $s$  を観測した結果が  $As = 0$  となりうる場合、 $As_0 = A(s + s_0)$  となる。 $s$  が  $K'$ -スパースであった場合、 $s + s_0$  は  $(K + K')$ -スパースとなるため、観測結果  $As$  からの  $l^1$  ノルム再構成が一意に定まらず、正しい推定結果を与えることは期待できない。そのため、スパースなベクトルに限った場合に線形変換  $A$  が元のベクトルの長さをあまり変えないことが、推定結果が正しいものとなるために望ましい。これを具体的に表現したものとして制限等長性 (Restricted Isometry Property; RIP) が定められており、任意の  $K$ -スパースな信号  $s$  に対して以下の式を満たす最小の  $\delta_K$  として定義されている。

$$1 - \delta_K \|v\|_2^2 \leq \|\Phi s\|_2^2 \leq 1 + \delta_K \|v\|_2^2 \quad (3.7)$$

観測行列  $A$  に対して、 $\delta_{2K} < \sqrt{2} - 1$  が成り立つような  $K \geq 1$  が存在するとき、任意の  $K$ -スパースなベクトルに対して  $l^1$  ノルム再構成は正しい結果を与えることが知られている。

### 3.3.6 復元誤差

ただしセンサからの信号は一般に完全にスパースな信号ではなく、いくらかの誤差を含む。しかし、もしスパースな表現においてエネルギーがごく少数の成分に集中している場合、スパース表現における零に近い成分は切り捨てて零とみなしても、大きな誤差とはならない。用いた基底行列が正規化されている行列である場合、スパース表現におけるごく少ないエネルギーを持つ成分を切り捨てても、基底行列を用いて元の信号に戻した際にそれが致命的な誤差には増加しないことが



(a) 周波数軸上における歩行中の加速度 (上位 20% 成分)  
 (b) 上位 20% 成分から復元した時間軸上における歩行中の加速度

図 3.3: 上位 20% のエネルギー成分による歩行中の加速度

保証されるためである．図 3.1(b) に示した歩行信号の加速度のスパース表現において，大きな絶対値を持つ上位 20% の成分のみを残し，残りの成分を切り捨てて零としたデータを図 3.3(a) に示す．そのデータに逆 DCT の基底変換を行って時間軸上の加速度データに戻したときのデータを図 3.3(b) に示す．赤い線が上位 20% 成分から復元したデータ，青い線が元のデータを表す．誤差は生まれるものの，わずかなエネルギーを持つ成分を切り捨てても元のデータを高い精度で復元できることが分かる．これにより，元の信号が完全にスパースな信号でなくノイズを含むような場合にも，復元の際にノルムを最小化するよう目的関数を定めることで，元の信号をよく再現できることが分かる．

### 3.4 行列の選択

前節までで，圧縮センシングの基本原則について説明した．本節では，本研究で取り扱う加速度センシングを行う参加型アプリケーションにおいて用いた観測行列と基底行列について説明する．

#### 3.4.1 観測行列 - sparse random matrix

本研究において用いた観測行列について説明する．Candès は各成分が一様分布やガウス分布に基づく乱数行列を，上述した RIP を満たす観測行列の例として挙げた．信号の圧縮処理はこの観測行列を加速度信号に掛けることによって行うため，携帯端末における圧縮処理をより軽量にするためには，よりスパースな行列を用いることが望ましい．

これに関連して Berinde らは，各行のランダムな  $d$  列に 1 成分を含み，他の成分は零であるような sparse random matrix を観測行列に用いることを提案している．こうした行列を用いても復元の際の誤差が悪くならず，RIP に近い性質が得られることを示した [38]．この  $m \times n$  の観測行列を元のデータ  $x \in \mathbb{R}^n$  にかける圧縮処理は， $x$  のランダムな  $d$  列を抜き出して和を計算することを  $m$  回繰り返すことに等しい．そのため，圧縮にかかる計算は  $d \times m$  回の足し算のみであり，非常に軽量に圧縮を行うことができる．そこで本研究においてはこの行列 ( $d = 4$ ) を用いて加速度信号の圧縮を行うものとした．

### 3.4.2 基底行列 - K-SVD

本研究において用いた基底行列について説明する．圧縮センシングにおいては何らかの基底上で信号がスパースに表現できることが望ましいが，どのような基底上で信号がスパースに表現できるかは信号の種類に依存し，未知なものである．そのため多くの圧縮センシングの利用法においては，自然界の信号や画像をある程度スパースに表現できることがよく知られている正規直交基底であるフーリエ基底やウェーブレット基底を用いることがほとんどであった．しかし近年，正規直交系ではなく冗長な表現を含む，つまり過完備 (Overcomplete) な基底行列を用いる試みが注目されている．センサからの信号は正規直交基底よりも過完備な基底を用いた場合の方がよりスパースな信号として表現できることが少なくないからである [39]．

それに関連して Ahron らは，何らかの訓練データを与えたときに，特異値分解 (Singular Value Decomposition; SVD) を利用して，それらをスパースに表現する基底行列を構築するコンパクトな行列を生成する K-SVD という手法を提案している [40]．本研究ではこれに注目し，取り扱う人間の行動の加速度信号に焦点を絞ることで，K-SVD と大規模な加速度データベースを用いて加速度信号に最適化した基底行列を構築し，フーリエ基底やウェーブレット基底よりもスパースに信号を表し，圧縮センシングを適用したときの復元誤差を低減させることを提案した．次節では K-SVD のアルゴリズムについて説明する．

### 3.4.3 K-SVD のアルゴリズム

$N$  個の訓練用サンプルデータ  $y_i$  を各列に含む行列を  $Y = (y_1, y_2, \dots, y_N) \in \mathbb{R}^{n \times N}$ ， $Y$  を近似的にスパースに表現する基底行列を  $D = (d_1, d_2, \dots, d_K) \in \mathbb{R}^{n \times K}$ ，また  $D$  上での  $Y$  のスパースな表現である係数行列を  $X = (x_1, x_2, \dots, x_N) \in \mathbb{R}^{K \times N}$  としたとき，K-SVD における目的関数は以下のように表される．

$$\min_{D, X} \{\|Y - DX\|_F^2\} \quad \text{subject to} \quad \forall i, \|x_i\|_0 \leq T_0 \quad (3.8)$$

ただし， $\|\cdot\|_F$  は  $\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}$  で表されるフロベニウスノルムである．

なお，K-SVD という名称は代表的なクラスタリング手法である K-means にちなんでつけられたものである．K-means ではあるベクトル集合が与えられたときに，各ベクトルが属する  $K$  個のクラスタをまずランダムに初期化する．その後，クラスタの重心位置を用いて各ベクトルが属するクラスタを更新する．同時に，新しくできたクラスタの重心を平均を用いて計算しなおす．この作業を繰り返すことでクラスタリングを行っている．K-SVD はその一般化を目指したアルゴリズムであり，K-means と同じく基底群  $D$  によるデータ行列  $Y$  の近似を行う初期化のフェーズと，基底群の更新を繰り返し行う反復処理のフェーズの2つからなる．基底群の更新を行う繰り返し段階においては，K-SVD は特異値分解を用いることで，各ベクトルがどの基底の線形和で表されるかを更新すると同時に，元のベクトルをよく表すように基底も更新することができる．各段階を順序を追って説明する．

#### (初期化) 基底群によるデータの近似

まず，基底行列  $D$  を初期化する． $D$  に訓練データ  $Y$  からランダムな  $K$  列を抜き出し， $D$  の各列にセットする．その後， $D$  の各列は  $l^2$  ノルムのもとで正規化する．

(反復処理) 特異値分解 (SVD) による誤差最小化

基底行列  $D$  の初期化を行ったあとは、スパース係数行列  $X$  の初期化を求めるため、目的関数 (3.8) を以下のような  $N$  個の目的関数に分解する。

$$\min_{\mathbf{x}_i} \{ \|\mathbf{y}_i - D\mathbf{x}_i\|_2^2 \} \quad \text{subject to} \quad \|\mathbf{x}_i\|_0 \leq T_0$$

$$\text{for } i = 1, 2, \dots, N. \quad (3.9)$$

$T_0$  が小さいときには、この問題は圧縮センシングの復元アルゴリズムと同じく基底追跡法などを用いることでよい近似解を求められることが知られている。これら  $N$  個の解を計算することで、 $X$  が得られる。

$D$  と  $X$  が得られた後は、これらを更新する。まず、 $D$  のうち 1 列、 $X$  のうち 1 行のみを残して他の成分を固定する。 $D$  の第  $k$  列を  $d_j$ 、 $x$  の第  $j$  行を  $x_T^j$  ( $j$  列ではなく、 $j$  行であることに注意が必要である) としたとき、 $D$  と  $X$  においてこれらの成分のみを変化させることを考える。このとき、上述した目的関数 (式 3.8) のペナルティ項は次のように表すことができる。

$$\begin{aligned} \|Y - DX\|_F^2 &= \left\| Y - \sum_{j=1}^K d_j x_T^j \right\|_F^2 \\ &= \left\| \left( Y - \sum_{j \neq k} d_j x_T^j \right) - d_k x_T^k \right\|_F^2 \\ &= \|E_k - d_k x_T^k\|_F^2 \end{aligned} \quad (3.10)$$

$DX$  を  $K$  個の行列に分解し、そのうち  $K-1$  個を固定して  $k$  番目の行列のみを変化させることを意味する。 $E_k$  は  $d_j$  を近似に用いなかったときの誤差を表す。 $E_k$  は固定されているため、この残差を最小化するためには、適当な  $d_k$  と  $x_T^k$  を求める方法が必要である。そこで、 $E_k$  を特異値分解 (singular value decomposition, SVD) を用いることが考えられる。ある行列  $M$  が与えられたとき、 $M = U\Delta V^T$  となる  $M$  の分解が存在し、これを  $M$  の特異値分解と言う。誤差を表す行列  $E_k$  を特異値分解した結果を  $d_k$  と  $x_T^k$  に代入することで、誤差を最小化する基底とその係数である  $d_k$  と  $x_T^k$  を同時に得ることができる。しかしながら、このまま特異値分解を用いた場合には、スパース性に関する制約が無い場合  $x_T^k$  は多くの非零成分で埋まってしまう。そこで、 $K$  列の  $Y$  のうち、近似のために  $d_k$  が使われている列番号を抜き出した集合  $\omega_k$  を導入する。

$$\omega_k = \{i \mid 1 \leq i \leq K, \quad x_T^k(i) \neq 0\} \quad (3.11)$$

$\Omega_k \in \mathbb{R}^{N \times |\omega_k|}$  を、 $(\omega_k(i), i)$  成分が 1 でそれ以外を零成分とする行列とし、 $x_R^k$ 、 $Y_k^R$ 、 $E_k^R$  をそれぞれ以下のように定義する。

$$\mathbf{x}_R^k = \mathbf{x}_T^k \Omega_k \quad (3.12)$$

$$\mathbf{Y}_k^R = \mathbf{Y} \Omega_k \quad (3.13)$$

$$\mathbf{E}_k^R = \mathbf{E}_k \Omega_k \quad (3.14)$$

つまり、 $\Omega_k$  を掛け合わせることによって、 $x_R^k$  は  $x_T^k$  における非零成分のみを取り出した  $|\omega_k|$  次元ベクトルとなる。また  $Y_k^R \in \mathbb{R}^{n \times |\omega_k|}$  は  $Y$  のうち  $d_k$  を近似の際に用いた列のみを取り出すことに相当する。同様に  $E_k^R \in \mathbb{R}^{n \times |\omega_k|}$  は、 $d_k$  を近似に用いたデータによる誤差を表す。

これらを用いて、再び目的関数のペナルティ項は以下のように表すことができる。

$$\|E_k \Omega_k - d_k x_T^k \Omega_k\|_F^2 = \|E_k^R - d_k x_R^k\|_F^2 \quad (3.15)$$

ここで  $E_k^R$  に特異値分解を行い  $E_k^R = U \Delta V^T$  とし、新しい基底として  $\tilde{d}_k$  に  $U$  の第1列を代入する。これは、特異値分解をした際に左特異ベクトルの第1列には、元の行列の分散を最大化する成分、すなわち第1主成分が得られることが知られており、誤差を最小化するための最も特徴的な量となるためである。その後、 $x_R^k$  には  $V$  の第1列に  $\Delta(1,1)$  を掛けたものを代入する。 $x_R^k$  には非零成分しか代入されていない状態であったため、この更新によって非零成分の数が増加してしまうことは起こらず、初期化されたときのスパース性の制約  $\|x_i\|_0 \leq T_0$  を満たすことができる。この処理を  $k = \{1, 2, \dots, K\}$  において反復し、収束条件を満たすまで繰り返す。

以上のアルゴリズムを簡潔にまとめるため、付録に K-SVD のアルゴリズムを表すチャートを示した。

#### 3.4.4 K-SVD を用いることのメリットとデメリット

圧縮センシングにおいて圧縮 (観測) 処理にはセンサデータに関する前知識が必要無く、ランダムに元データからサンプリングを行う観測行列を掛け合わせることによってのみ圧縮が終了する。その分、復元の際にはそのデータがスパースに表現できる基底が必要となる。正規直交性を持つ DCT や DWT を用いてもある程度のスパース性が得られることが知られているが、それはある種類のセンサデータに最適化されたものではない。人間の行動の加速度をセンシングすると決定している場合に、そのデータに最適化された基底行列が望ましい。K-SVD は訓練データをスパースに表す基底行列を構築する手法であり、我々は後述するように大規模な加速度データベースを利用可能である。これらを組み合わせて、人間行動の加速度データに対象を絞ることで、加速度データに対する圧縮センシングに最適な基底を作り出すことができる。

ただ、基底行列を構築する際には訓練データに使われなかった行動を端末の持ち主がとった場合に、K-SVD による基底行列が有効であるかどうかは定かではない。用いるアプリケーションを絞った上で基底行列を使うこと、もしくは他の行動が取られた場合に使用する基底行列を自動的に変更するような仕組みが必要であると考えられる。また、K-SVD により構築する基底行列は、 $K$  の値を大きく設定し、DCT 行列などの正方行列に比べて列数を非常に大きく設定した方が復元誤差が小さくなる場合がある。しかし、列数を増やすと圧縮センシングにおけるデータ復元にかかる計算量が増加する。本研究のような比較的次元の低い、加速度のようなデータを用いる場合にはさほど問題とはならないが、画像など高次元のデータに K-SVD による基底行列を適用する場合、復元にかかるリソースや時間に注意が必要であると考えられる。

### 3.5 参加型加速度センシングへの適用

Carroll らによると、通話、Web ブラウジング、メール、音楽再生など日常的な用途を想定した携帯電話において、完全に充電された状態からバッテリーを消耗しきるまでの 27 時間において消費する電力は通信による部分が一番大きく 44%程度、次いで CPU による部分が 14%程度であり、

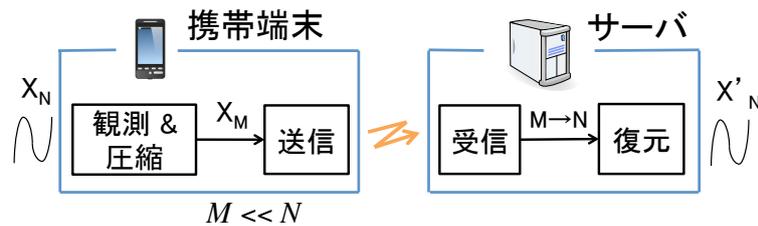


図 3.4: 圧縮センシングを用いた参加型センシング

残りはグラフィックやバックライトによる部分がほとんどである [41]。もちろんこれは携帯端末のスペックにも大きく左右される数値であるが、画面をスリープさせた状態でセンシングとデータ送信を連続的に行うアプリケーションを想定した場合に、通信と CPU による電力消費を考慮することが低消費電力なアプリケーションを開発する上で最も望ましい。

図 3.4 に、圧縮センシングを用いた場合の参加型センシングの構成図を示す。通信における消費電力を小さく抑えようとした場合、送信するセンサデータのサイズを小さくすることが必要になる。なるべく生データに近い形でデータの送信を行ってサーバ側でデータの保存や分析処理を行い、リアルタイムに参加者にフィードバックを行うことで時間や個人に特化したサービスを行う参加型センシングを考えた場合、センサデータは圧縮される必要がある。

しかし、既存の ZIP などによる情報圧縮を行った場合、圧縮処理に大きな計算量を用いるため CPU に負荷をかけ、逆に電力を消費することが考えられる。そこで、最小限の計算量で圧縮を行える圧縮センシングを携帯端末におけるセンシングに適用することで、復元を行うサーバ側には複雑な計算処理を必要とするものの、端末側の消費電力を低減できることが考えられる。次章からの評価実験により、この提案によって実際に携帯端末の消費電力を低減できることを示す。

## 第4章

---

### 復元誤差と行動認識率の評価

## 4.1 はじめに

本章では、これまでに述べた圧縮センシング、ならびに圧縮センシングに用いる観測行列と基底行列の構築方法に基づき、加速度データに圧縮センシングを適用したときに生じる復元誤差について評価する。NRMSEに基づく各行動の加速度データの誤差の評価に加え、実際にそれがデータ受信側、つまりサーバ側で行動認識に用いられたときにどのような影響を及ぼすかについて評価を行う。

## 4.2 データセット

### 4.2.1 HASC2010corpusの利用

前章で説明した K-SVD のアルゴリズムを適用するには訓練データが必要となる。筆者らは人間の行動の加速度データをスパースに表現するため、Human Activity Sensing Consortium (HASC) から提供される大規模加速度データベースである “HASC corpus” を用いた [42]。音声認識や画像認識の分野などでは大規模なコーパスが重要な役割を果たしており、Human Activity Sensing Consortium (HASC) では、センシングを通じて人間の行動を認識、理解することを目的に、加速度センサを含む装着型センサの大規模データベースである “HASC corpus” の構築を行っている。K-SVD に適用する訓練データには、その中から “HASC2010corpus” を使用した。このコーパスは人間の基本的な 6 行動 (静止, 歩行, 走行, スキップ, 階段昇降) とその全てを一度に行ったシーケンスデータからなっている。データは各実験協力者からボランティア的に集められたものであり、加速度データ本体だけでなく、どの行動を取っているかというラベル情報、サンプリング周波数、実験協力者の年齢や性別、データ取得時の床の材質、端末の保持位置、端末の機種など、細かいメタデータが付属されている。実験協力者はどのような端末を使ってもよいが、実際に販売され流通している端末であることが必要となっている。HASC2010corpus は合計 540 名の実験協力者のデータから構成されており、全ての行動が揃っているのはそのうち 96 名分である。加速度データファイルの総数は 6791 で、総データサイズは 966MB にのぼる。全ての実験協力者のデータについて全ての行動が揃っているわけではないため、本稿においては、全ての実験において HASC2010corpus の中から 90 名分の 100Hz のデータを選択して用いている。

また、もともと携帯端末に内蔵される加速度センサやジャイロは、携帯端末の姿勢を判別するために取り付けられたものであり、加速度センサでは 3 軸からの情報が得られる。こうしたセンサを行動認識に用いる研究の多くでは、複数の端末や基盤を使用したり、センサを体の特定の部位に固定することを制約として設けているものが多い。しかしながら、日常生活における携帯端末の保持位置は腰ポケット、胸ポケット、ハンドバッグの中など様々である。そうした日常生活においてセンシングを行えるシステムを想定したとき、携帯端末の保持位置や姿勢に関わらず認識を行えることが望ましい。そのため、本研究では、加速度センサの  $l^2$  ノルムを計算して、重力加速度の影響を取り除いて用いた。つまり圧縮前の元信号  $x$  は、3 軸加速度センサの値 ( $Acc_x, Acc_y, Acc_z$ ) を用いて  $x = \sqrt{Acc_x^2 + Acc_y^2 + Acc_z^2}$  として処理される。

図 4.1 に、HASC2010corpus に含まれる加速度データの例を示す。データは全て重力加速度  $9.81\text{m/s}^2$  を 1.0 として正規化されており、静止信号はほぼ 1 に近い値を示す。その他の行動は周期的な変動を示す。歩行信号の例では、加速度センサが左右どちらかの腰ポケットの中に固定されている場合が多いため、センサが挿入されているポケット側の足を踏んだ瞬間に大きな加速度の値を示し、逆側の足を踏んだ瞬間もある程度大きなピーク値を示す。走行信号やスキップ信号は歩行

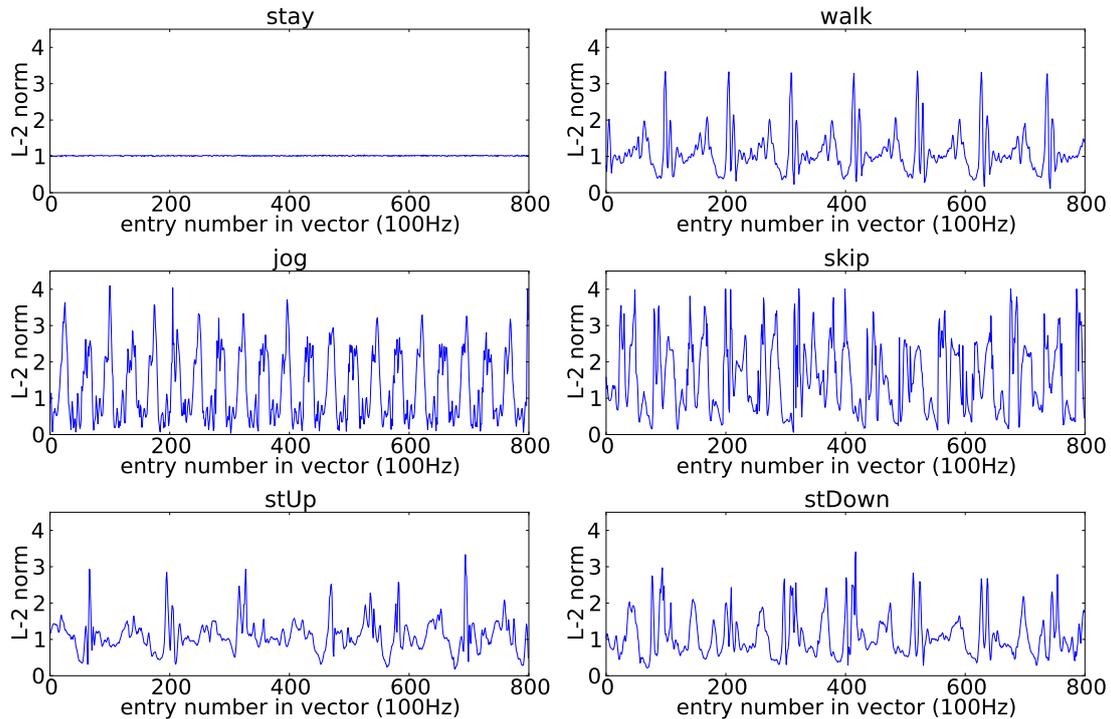


図 4.1: HASC2010corpus に含まれる行動の加速度データの例

信号に比べて行動の周期が短く、また大きな加速度の値を示す。階段昇降信号は 2 者の中で区別が難しいとともに、歩行信号とも酷似している。

#### 4.2.2 データの前処理

HASC corpus は協力者からボランティア的に集められたデータであり、データの長さもまちまちである。使用する 90 人分、6 行動、各行動 4 セット、100Hz のデータは最短で 12 秒程度である。データの最初と最後には端末をポケットに固定している動作、またポケットから取り出してセンシング終了ボタンを押している動作などが含まれていることがあるため、データ開始から 2.0 秒間が経過してからの 8.0 秒間のデータを切り取るようにオフセット値を設定した。さらに、8.0 秒間の全データを  $2.0 \times 4$  つの窓に分割した。よって、用いられたデータは  $90 \times 6 \times 4 \times 4 = 8640 \times 2.0$  秒間のデータである。

#### 4.2.3 K-SVD による基底行列の構築

復元誤差と行動認識への影響を評価するにあたり、圧縮センシングを用いて上述したデータセットを圧縮・復元したデータを用意する必要がある。その際に用いる基底行列を K-SVD によって構築する方法について説明する。

まず、10 分割交差検定を行うため、90 人分の加速度データのうち 81 人分を用いて基底行列を構築に用いる。合計して、基底行列の構築のために、6 つの行動がそれぞれ  $81 \times 4 \times 4 = 1296$  ( $\times 100$  Hz, 2.0 s) のデータを持つことになる。これらの各行動のデータを  $Y_i \in \mathbb{R}^{200 \times 1296}$  ( $i = 1, 2, \dots, 6$ ) として

K-SVD の訓練データに用いた．まず，各行動に最適化された基底行列を作成するため， $Y_i$  のそれぞれに K-SVD を適用して 6 つの基底行列  $D_i$  ( $i = 1, 2, \dots, 6$ )  $\in \mathbb{R}^{200 \times 300}$  を作成する．このときの K-SVD のパラメータは  $K = 300, T_0 = 50, i$ (繰り返し回数) = 10 とした．そしてこれらの基底行列を水平に連結した行列  $D$  を，圧縮センシングに用いる最終的な基底行列とした．

$$D = [D_1 \ D_2 \ \dots \ D_6] \in \mathbb{R}^{200 \times 1800} \quad (4.1)$$

正確には，この基底行列に，全ての成分が 1.0 からなる 200 次元の列ベクトルを追加する．これは，K-SVD は特異値分解によってある行動において異なるデータ間の差を最もよく表すような主成分を抜き出して基底行列に加えようとするため，静止信号のように変動が小さく個人差の出にくい信号のみを用いて作られた基底行列においては，直線的な信号を表す成分が含まれておらず復元がしにくくなることが懸念されたためである．

なお，K-SVD の実装には MATLAB 用に開発された KSVDS-BOX を用いている [43]．その他の行列操作なども含め，MATLAB 上で処理を行った．

#### 4.2.4 圧縮・復元を行ったデータセット

また，圧縮率による誤差や行動認識率への影響を調べるため，上述した 90 人分の全てのデータに対して 5% から 100% まで圧縮率を変化させながら圧縮センシングを行って復元した 20 個のデータセットを用意した．元データの窓幅は  $2.0 \text{ s} \times 100 \text{ Hz} = 200$  次元であるので，観測行列には各列に 4 つの 1 成分を含む 200 列の sparse random matrix を用いている．行数は圧縮率によって定まるため，5% から 100% の圧縮率に応じて 10 行から 200 行まで変動する．その際使用する基底行列には，K-SVD を利用した基底行列と，比較対象として従来圧縮センシングにおいてよく使われてきた DCT 行列を用いた．

### 4.3 復元誤差の評価

#### 4.3.1 NRMSE による交差検定

以上の方法で構築した基底行列，ならびに sparse random matrix による観測行列を用いて圧縮センシングを行い復元されたデータに対して，次式で表される Normalized Root Mean Square Error (NRMSE) による評価を行った．評価は 10 分割交差検定を用いて行っている．つまり，81 人分のデータを用いて基底行列を構築し，残りの 9 人分のデータに圧縮センシングを用いて復元したデータに対して NRMSE を算出する処理を 10 回繰り返して平均によって評価した．

$$NRMSE(\hat{x}, x) = \frac{\sqrt{\frac{\sum_{i=1}^n (\hat{x}_i - x_i)^2}{n-1}}}{x_{max} - x_{min}} \quad (4.2)$$

#### 4.3.2 復元の例

実際に圧縮センシングを行い，様々な圧縮率で圧縮・復元を行った歩行信号の例を図 4.3 に示す．図 4.3 において赤線が復元した信号，青線が元の信号である．30% の圧縮率からでも，細かいピークを捉えて元の信号をよく復元できていることが分かる．

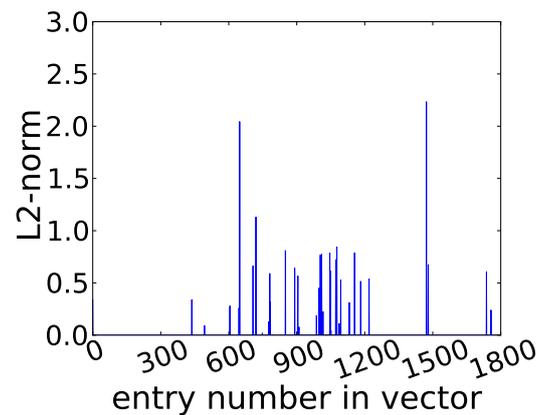


図 4.2: 圧縮率 30%において復元した歩行信号のスパース表現

また，図 4.2 に，歩行信号の用いた K-SVD による基底行列のうち何列目の成分を使うことで復元を行っているかを示すスパース表現を示す．前述した基底行列の構築方法に従い，1 列目が全ての値が 1.0 からなる信号，2 から 301 列目は静止信号，302 から 601 列目は歩行信号，602 から 901 列目が走行信号，902 から 1201 列目がスキップ信号，1202 から 1501 列目が階段を昇る信号，1502 から 1801 列目が階段を降りる信号に対応している．歩行信号は階段昇降など他の信号に酷似していることも多く，この例では歩行信号を復元するために階段昇降だけでなく走行やスキップの信号から抜き出した成分も用いていることが分かる．

### 4.3.3 結果

圧縮センシングを行ったデータに対して，行動ごとに復元誤差を算出した結果を図 4.4 に示す．

DCT 行列を用いた場合と比較して，静止信号を除いた 5 つの行動で復元誤差が減少していることが分かる．例として歩行信号の圧縮率 40% の点において，DCT 行列を用いたときは  $\text{NRMSE}=0.1042$ ，K-SVD による基底行列を用いたときは  $\text{NRMSE}=0.0574$  である．歩行信号においては K-SVD による基底行列を用いることで，DCT 行列を用いる場合と比較して約 45% の誤差を削減できることが分かる．違う言い方をすれば，DCT 行列と同じだけの誤差を許すとき，K-SVD を用いることで圧縮率を 20% 程度下げることができる．

ただ，静止信号は他の信号と比べて復元誤差が大きく，DCT 行列のそれよりも大きな誤差を生じる．これは静止信号自体が変動が少なく振幅の小さなものであるため，正規化した誤差 (NRMSE) では大きな数値として出るためであり，復元において信号の概形を損ねるものではない．正規化しない RMSE で比較すると DCT 行列を用いた場合とほとんど変わらない．次節で説明する行動認識の結果からも，静止信号の誤差は問題とならないことが分かる．

## 4.4 行動認識率の評価

これまでで，加速度データに圧縮・復元を行ったときの誤差について評価した．復元誤差は K-SVD を用いたことで小さく抑えられることが分かったが，こうした復元誤差が実際のアプリケーションにおける加速度データの用途においてどのような影響を及ぼすのかは明らかでない．HASC2010 corpus

に含まれるような行動を認識する典型的なアプリケーションにおいて、復元誤差が行動の認識率に大きな影響を及ぼすかどうかを調べた。

#### 4.4.1 加速度を用いた行動認識の研究例

体に取り付けた加速度センサのデータから行動を認識する試みは2000年頃からなされ、既に多くの研究が行われ成熟しつつある。ここでは、加速度データからの行動認識の代表的な研究について述べる。Baoらの研究では、体の5カ所に取り付けた2軸加速度センサからのデータから、平均や加速度データ間の相関などを特徴量として抽出することで、最高で84%の精度で日常的な20行動を認識できることを示しており、また特定のユーザの加速度データを本人のデータのみを用いて訓練した方が高い認識率を出すことができることなどを説明している [44]。

Lesterらの研究では、体のある一部分にのみ加速度センサやマイクなど8種類のセンサが組み込まれた端末を取り付け、8種類の行動を約90%の精度で認識できることなどを示した [45]。また、12人の実験協力者のデータを用いて、訓練に用いた人数が多いほど認識精度が高くなることも示している。

近年こうした認識手法は全てスマートフォンを始めとする携帯端末のみを用いて行えるようになってきており、行動認識を行うために専用のセンサを体に取り付けるという負担を無くすことができる。Reddyらは携帯端末のみを用いて、GPSから計算される移動スピードと加速度データのノルムの分散ならびに離散フーリエ変換を行ったときの1-3Hzの成分をDiscrete Hidden Markov Modelに用いることで、歩行、走行、自転車走行、電車による移動を約94%の精度で認識することに成功している [46]。

#### 4.4.2 本研究における行動の認識手法

事例によってセンサの数、位置、種類が様々であり、センサが固定されているかどうか、どういった環境で測定を行ったか(屋内、屋外など)によっても精度が変わってくる。認識すべき行動の種類が増えることによっても、精度の数値は大きく変わってくるため、どの事例における手法が最適であるかを判断することは難しい。

本研究では日常的な用途を想定し、1つの3軸加速度センサを備えた携帯端末1台のみを用いることにする。また携帯端末の姿勢や位置を考慮せずに認識が行えるようにするため、加速度のノルムのみを用いている。他の電力を消費する要素や認識率に影響する要素を排除するため、行動認識には加速度データのみを用い、GPSなど他のセンサ情報を用いないこととしている。

加速度データに圧縮センシングを用いることで復元した際の誤差を小さく抑えられることが分かったが、こうした誤差が行動認識など代表的な加速度の利用方法に用いられたとき、精度に大きな悪影響を与えることが懸念される。そうした復元誤差の行動認識への影響を調べるために、実際に機械学習を加速度データに用いて認識率を評価した。行動の認識方法として、センサの数が1台であること、加速度データのみから認識を行うことなどから、Kwapiszらによる方法に基づいて以下の特徴量を抽出している。

- 平均
- 分散
- 零点交差率 (Zero Crossing Rate; ZCR)

- 平均絶対誤差 (Average Absolute Difference; AAD)
- 10 Binned Distribution
- 周波数帯ごとのエネルギー (0.5-1.0 Hz, 1.0-1.5 Hz, 1.5-2.0 Hz, 2.0-2.5 Hz, 2.5-3.0 Hz)

零点交差率は、加速度データの  $l^2$  ノルムが重力成分と同じ 1.0 であるときを零点として、単位時間あたりに零点と交差する回数として計算している。10 Binned Distribution とは、1 サンプル中の加速度ノルムの (最大値 - 最小値) の範囲を 10 分割し、サンプル中の加速度の値がどの範囲に属するかの割合を示した 10 個の特徴量である。エネルギーに用いる周波数帯には、Reddy らの方法などを元に、行動によって差が出やすいとされている周波数帯を選択している [12]。

加速度ノルムに適用できる特徴量として、Kwapisz らはこの他にピーク間隔 (Time Between Peaks) を用いているが、この算出はヒューリスティクスに定められた閾値にのみ基づいてピークを定義しており、また静止などのピークを持たない行動に対しては適用しにくいいため、本研究では用いていない。

分類器には Kwapisz らの研究を参考に、機械学習用の総合ツールキットである Weka に含まれる J48 決定木を用いた [23]。Weka には簡単に扱える GUI も用意されているが、本研究では、その他に提供されている Java 向けのライブラリを Eclipse 3.6 (Helios) から用いている。GUI でできることはライブラリに含まれている機能のごく一部であることに加えて、圧縮率を変更させながらの認識の繰り返しなどを行う場合、ライブラリを用いる方が容易にシミュレーションが行える。なお、J48 決定木の実行時オプション値には“-C 0.25 -M 2” (枝刈りの強度が 0.25, 1 つの葉に属する最小の事例数が 2 であることを意味する) を用いている。

使用したデータは、上述した HASC2010corpus の全データを 5%刻みに圧縮率を変化させて復元したものである。認識精度の算出には 10 分割交差検定を用いており、学習用データセットに含まれる実験協力者とテスト用データセットに含まれる実験協力者は別の人間になるように設定し、ユーザ独立認識を行っている。また、圧縮と復元を行ったデータセットの行動認識率をテストするときにも、圧縮を行っていない元のデータを用いて学習した分類器を用いて認識を行った。

#### 4.4.3 結果

まず、図 4.1 に圧縮を行わない元データを用いたときの認識率を表す confusion matrix を示す。全体で、認識精度は 76.9%であった。分類ミスのほとんどは歩行と階段昇降の分類ミスに起因していた。歩行と階段昇降は加速度データが類似していて、図 4.4 にも見られるように復元誤差も似た変化をしており、区別が難しい。また、単純に歩行と階段昇降の際の体の動きが似ていることから加速度ノルムに差が現れにくいことに加えて、階段昇降の途中に踊り場で歩行を挟んでいるデータが含まれているなど、HASC2010corpus 自身にある問題も関連していると考えられる。また、HASC2010corpus には、端末がハンドバッグに入っている状態や、固定されていないジャケットのポケットに入っている状態で観測された加速度データも含まれている。こうしたデータは腰ポケットに固定されている場合と比べて加速度の変化が小さく認識率は低くなるため、全体の認識精度にも影響を及ぼすと考えられる。

次に、図 4.5 に、各行動ごとの圧縮率と認識精度との関係を示す。圧縮率 30%程度までは、どの行動に関しても認識率をほとんど落とさずに行動認識を行うことができる。圧縮率 30%における全体の認識率は 73.0%であり、圧縮を行っていないデータを用いたときと比べても 4%しか低下していない。

表 4.1: 非圧縮の加速度データを用いて行動認識を行ったときの認識精度 (Overall: 76.9%)

|               |        | Predicted Class |            |             |             |            |            | Recall (%) |
|---------------|--------|-----------------|------------|-------------|-------------|------------|------------|------------|
|               |        | Stay            | Walk       | Jog         | Skip        | StUp       | StDown     |            |
| Actual Class  | Stay   | <b>1413</b>     | 27         | 0           | 0           | 0          | 0          | 98.1       |
|               | Walk   | 4               | <b>839</b> | 9           | 5           | 320        | 240        | 59.2       |
|               | Jog    | 0               | 13         | <b>1291</b> | 105         | 3          | 24         | 89.9       |
|               | Skip   | 0               | 7          | 101         | <b>1301</b> | 11         | 20         | 90.3       |
|               | StUp   | 0               | 309        | 0           | 7           | <b>892</b> | 232        | 61.9       |
|               | StDown | 0               | 240        | 21          | 27          | 245        | <b>907</b> | 63.0       |
| Precision (%) |        | 99.7            | 58.5       | 90.8        | 90.0        | 60.6       | 63.7       |            |

圧縮率 30%を下回ってからの認識率については、行動ごとに認識率の異なる変化が見られた。まず静止状態はその信号自体にほとんど変化が無いので、どんなに圧縮率を高めても信号の概形が壊れることはなく、認識精度もほとんど変わらない。他の行動については、認識に用いる特徴量セットが行動ごとに異なるために認識精度に差が生まれている。

走行、スキップといった行動は圧縮率 30%程度までは他の行動に比べて比較的高い認識率を保っているが、それ以降は急激に認識率が低下することが分かる。これらの行動は他の行動に比べて運動量が大きく、はっきりとした強弱を持った加速度が見られるが、その分速度や足を出すリズムなどに個人差が出やすい。そのため K-SVD による基底行列を用いて復元を行うことが他の行動に比べて難しく、NRMSE で評価したときにも歩行信号に比べて同じ圧縮率で比較したとき大きな誤差を持つことが図 4.4 にも示されている。

また、上述した歩行信号と階段昇降の信号の分類ミスは圧縮率が高くなるほど顕著に見られる。圧縮率が 20%のときには、階段昇降信号のうち 33.5%は歩行信号と分類されてしまっている。収集した加速度データをカロリーメータのような e-Health アプリケーションに用いる場合、こうした行動を区別することは重要な情報である。圧縮率を更に高めても行動をよく分類したいとき、こうした歩行と階段昇降を上手く分類する分類器、特徴量を見つけることが必要である。

#### 4.4.4 特徴量の値の変化

圧縮が行動認識に与える影響の原因を調べるため、認識に用いた特徴量の値が圧縮によってどのように変化してしまうのかを調べた。用いた 19 個の特徴量のうち、平均、分散、Binned Distribution 5/10、周波数帯ごとのエネルギー (2.0-2.5 Hz) の値の変化を示す。平均値は運動の強度を表す指標であり、行動認識において重要な特徴量である。圧縮を行っても平均値はほとんど変化せず、圧縮に強い特徴量であることが分かった。その他の特徴量は、圧縮率によって値が大きく変化する。歩行信号 (緑線) に比べて走行やスキップは圧縮により大きく値が変化することが多く、こうした変化が図 4.5 における認識率の変化の違いの要因となっている。こうした特徴量の変化の性質を調べることで、圧縮率が高いときにどの特徴量を用いた方が信頼性が高いかといった最適化を行える可能性がある。

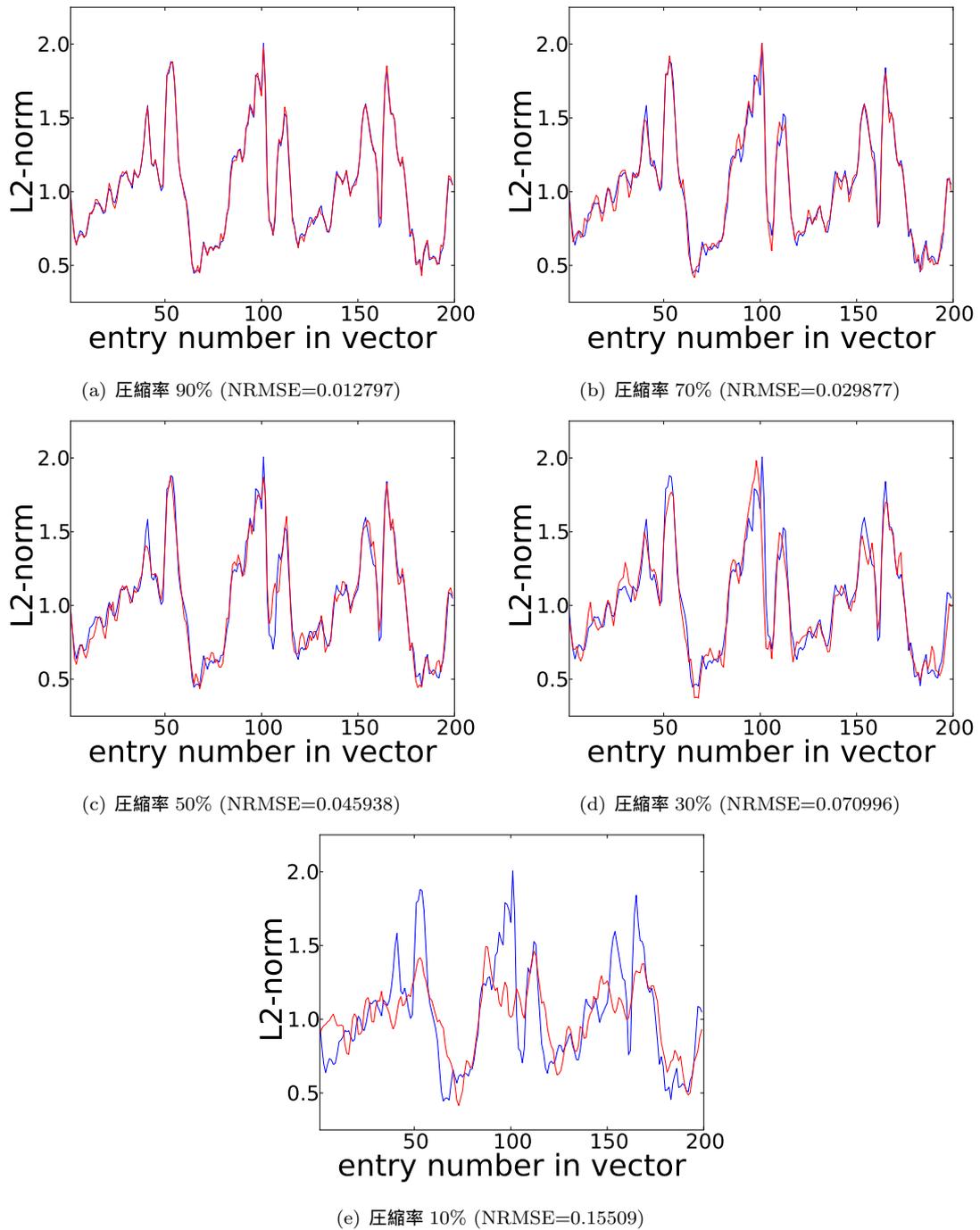


図 4.3: 様々な圧縮率で復元した歩行信号

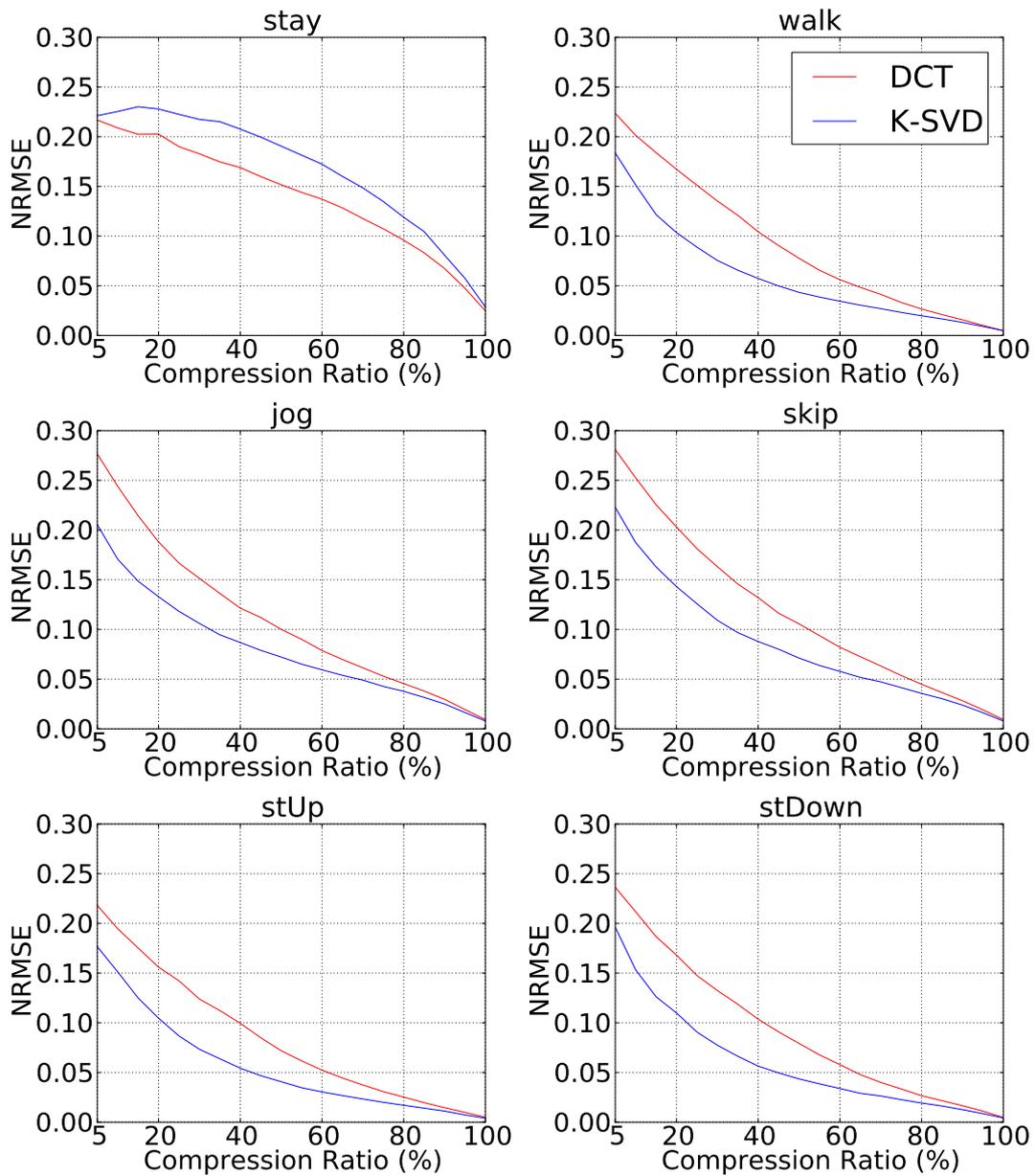


図 4.4: { 静止, 歩行, 走行, スキップ, 階段昇降 } の 6 行動を K-SVD により構築した行列を用いて復元したときの平均誤差と圧縮率の関係

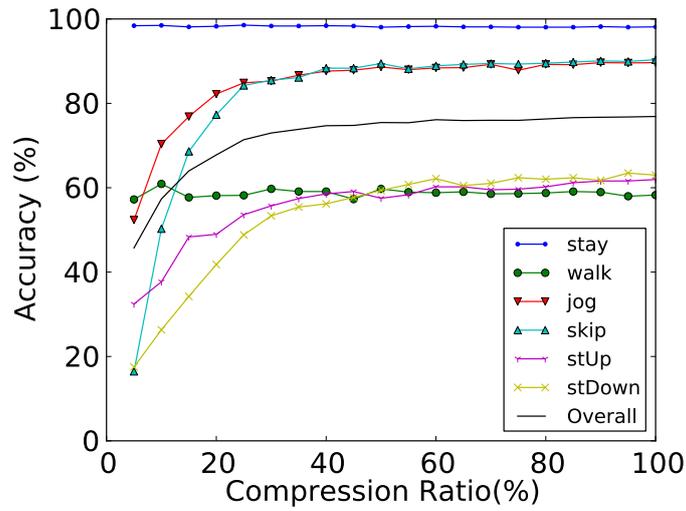


図 4.5: 圧縮率と認識精度の関係

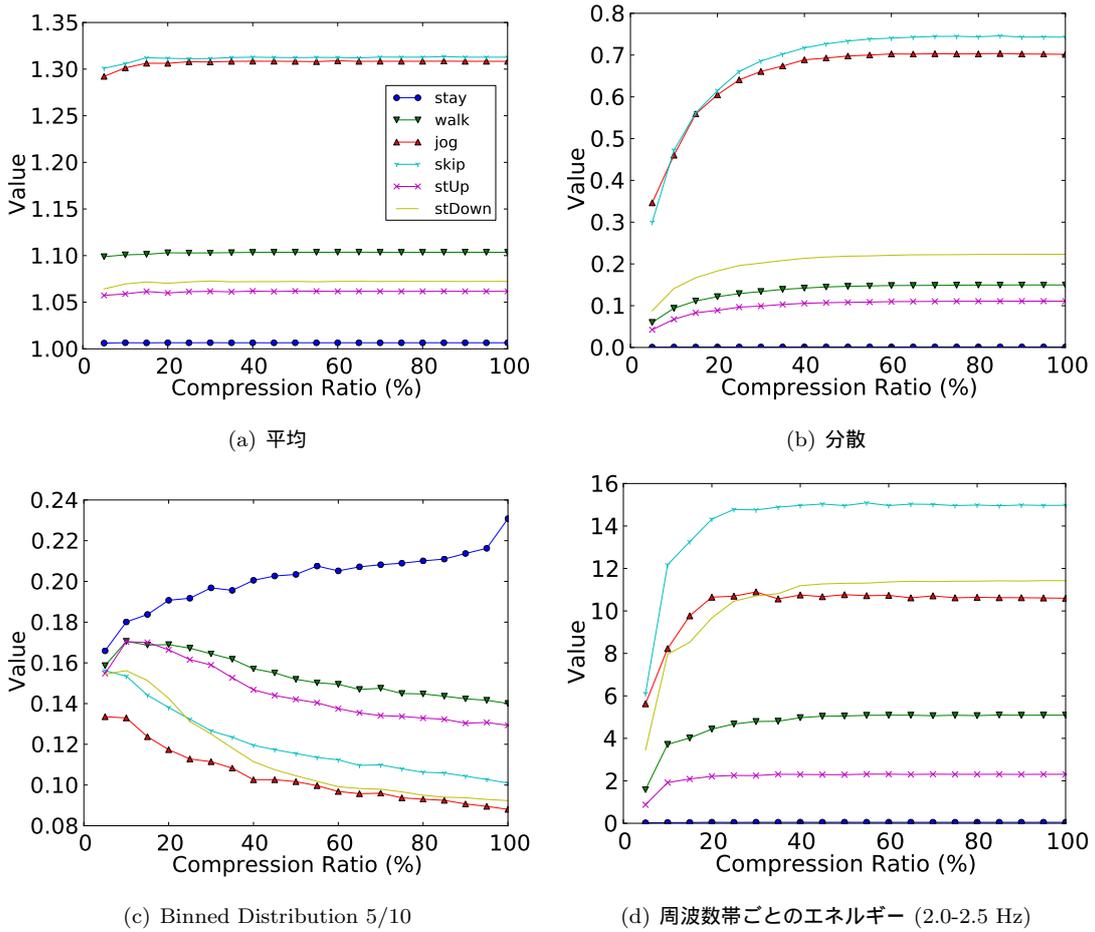


図 4.6: 行動認識に用いた特徴量の値と圧縮率の関係

## 第5章

---

### 消費電力の評価

## 5.1 はじめに

本章では、提案した圧縮センシング手法を実際の Android 端末に実装したときに消費する電力について明らかにする。また、典型的な情報圧縮として ZIP による圧縮と比較して、提案手法がどの程度消費電力が削減できるのかを明らかにする。まず、実際の Android 端末を用いて消費電力を測るときに必要な機器やハードウェア上での準備と、消費電力を測るアプリケーションの準備について説明する。その後、センシングにかかる電力、データ圧縮にかかる電力、データ送信にかかる電力など処理ごとの消費電力を明らかにする。最後に、それらを組み合わせた全体での消費電力を比較し、圧縮センシングを用いることでどのように消費電力が低減できるかを説明する。

## 5.2 実験概要

本研究における提案手法の省電力性を評価するため、実際の携帯端末を用いてその消費電力を計測した。携帯端末において、本研究で提案するセンシングアプリケーションを駆動することによって電力を消費する要素は主に以下の 3 つの処理である。

- 加速度センシング
- データ圧縮
- データ送信

これら単体の消費電力を測定し、次にこれら全部を行ったときの消費電力を測定した。

### 5.2.1 各処理の消費電力

#### 加速度センシング

加速度センシングのサンプリングスピードは現在の Android SDK において `SensorManager.SENSING_DELAY_{NORMAL, UI, GAME, FASTEST}` の 4 種類が提供されている。これらはサンプリング間隔の目安を示すだけのものであり、実際にこれらが何 Hz の周波数となるかは端末のスペックとアプリケーションの処理によって変わってくる。用いた Android 端末におけるこれらの周波数と消費電力を調べた。

#### データ圧縮

データ圧縮においては、評価として、

- 圧縮センシングによって圧縮率を {10, 30, 50, 70, 90}% に変化させながらそれぞれ圧縮を行う。用いる観測行列は各行のランダムな 4 列に 1、それ以外が 0 で構成される sparse random matrix を用いる。
- 典型的な情報圧縮方法として、ZIP による圧縮を行う。

の方法を比較した。圧縮するデータサイズを {10, 50, 100, 500, 1000} kbyte で変化させながら、消費電力がどのように増加するかを測定した。なお、圧縮センシングを用いるときには、観測行列の列数は200列としている。そのため一度に圧縮できるのは200次元の加速度データであり、それ以上のサイズのデータの圧縮は、これを繰り返すことを行っている。観測行列のサイズを大きくするとそれだけ観測行列の保存に必要なメモリが大きくなるため、一度に大量のデータを圧縮する観測行列を用意することは望ましくない。

### データ送信

センシングした加速度データを遠隔のサーバに送信する際の消費電力を計測する。屋外において日常的にセンサデータを送信する場面を想定して、3G ネットワークを用いて HTTP POST メソッドでデータを送信したときの電力を計測した。b-mobile 3G データ通信用プリペイド SIM カードを用いており、docomo のネットワークを利用していることになる。送信するデータサイズを {10, 50, 100, 500, 1000} kbyte で変化させながら、消費電力がどのように増加するかを測定した。

## 5.2.2 全体の消費電力

想定する参加型センシングアプリケーションは、基本的に以下の処理を繰り返す。

1. 加速度センシングを行い、データを保存する。
2. 一定時間経過後、加速度データを圧縮する。
3. データ圧縮が終わると、データをサーバに送信する。

これらの処理を1周期とし、1周期における端末の消費電力を評価した。そのとき、加速度センサのサンプリングスピードは

`SensorManager.SENSING_DELAY_FASTEST`

とした。ランダムな加速度データに ZIP による圧縮を行ったとき圧縮率は50%程度であったため、全体の比較では圧縮センシングにおける観測行列は100行200列として圧縮率を50%に揃えて行った。そのため、データ送信にかかる消費電力は2つの圧縮方法で同じとなる。また、加速度センシングを始めてから送信するまでの時間は300秒としている。

## 5.3 実験準備

### 5.3.1 ハードウェア・セットアップ

まず、消費電力を測るために用いた機器について説明する。携帯端末として、2011年12月2日に発売された、docomo GALAXY NEXUS SC-04D を用いた。主要なスペックを表5.1に示す [48]。端末には3G 及び Wi-Fi による通信機能が備わっており、また3軸加速度センサも内蔵されている。

この Android 端末の消費電力を計測するため、Thiagarajan らの方法を参考に、バッテリーを外部に取り外し、バッテリーから本体に流れる電流を計算した [49]。図5.1に用いた機器を示す。電流計測には Agilent 34410A デジタルマルチメータを用い、端末とバッテリーの間に0.1Ωの抵抗を直接に接続して抵抗間の電圧降下を測ることで電流値を計算した。そのときの電圧の値は、バッ

表 5.1: docomo GALAXY NEXUS SC-04D の主要スペック [48]

| Component | Specification                    |
|-----------|----------------------------------|
| CPU       | OMAP4460 1.2GHz (Dual Core)      |
| 発売時の OS   | Android 4.0 (Ice Cream Sandwich) |
| ディスプレイ    |                                  |
| 種類        | HD 有機 EL                         |
| サイズ       | 約 4.7 インチ                        |
| 解像度       | 720 × 1280 ドット                   |
| メモリ       |                                  |
| ROM       | 16GB                             |
| RAM       | 1GB                              |
| バッテリー容量   | 1730mAh                          |

バッテリーの電圧の計測により 3.7 V としている．バッテリーの残量による電圧の変化を考慮して，常に残量が約 50% の状態で消費電力計測の実験を行うようにした．

バッテリーを本体から取り外した状態で導線とともに強く固定する必要があったため，もう一台の GALAXY NEXUS を用意して，そのバッテリー保管用のスペースに図のように挟んだ状態にしてある．図で右側はケースとして使用しているのみである．この状態で左側の GALAXY NEXUS において様々なアプリケーションを起動し，電力の時間変化の積分値を計算することで消費電力量を計算した．全計測において電力のサンプリング間隔は 10 ms に設定している．

### 5.3.2 ソフトウェア・セットアップ

まず，各処理の消費電力を明らかにするため，Android 上において，加速度センシングのみを行い続けるアプリケーション，事前に用意して端末内に保存しておいた加速度センサデータを圧縮センシングを用いて圧縮するアプリケーション，同じく加速度センサデータを 3G ネットワークを通じて HTTP POST メソッドを用いて送信するアプリケーションを実装した．

また，携帯端末においてグラフィックとバックライトが占める消費電力は非常に大きい．センシングに関わる処理にかかる消費電力のみを計算する際には，そうしたディスプレイにかかる消費電力を排除して考えることが望ましいため，実験中はディスプレイをオフにした状態で行った．何の処理も行わずにアプリケーション駆動中にディスプレイをオフにした場合，CPU や通信も同時にオフになってしまうため，センシング処理も持続できない．そこで，Android SDK に含まれる *Wakelock* の機能を用いて，他のコンポーネントに影響を及ぼさないようにディスプレイをオフにした．具体的には，アプリケーション起動直後に *PARTIAL\_WAKE\_LOCK* を起動し，手動でディスプレイをオフにしても他の処理は継続されるように実装している．

Android はバックグラウンドにおいて駆動するアプリケーションが多く，バックグラウンドで通信を行うことで消費電力に大きな影響を及ぼすことがある．そうしたものを遮断するため，Google Play にて無料で配布されている “Advanced Task Killer” のアプリケーションを利用して，計測の直前に必ずバックグラウンドタスクを全て終了させてから行うようにした．

また，センサデータを受信するサーバも遠隔の PC 上に実装した．実装には Python 用のウェブサーバ構築用フレームワークである *web.py* を使用し，HTTP POST メソッドを用いて送信され

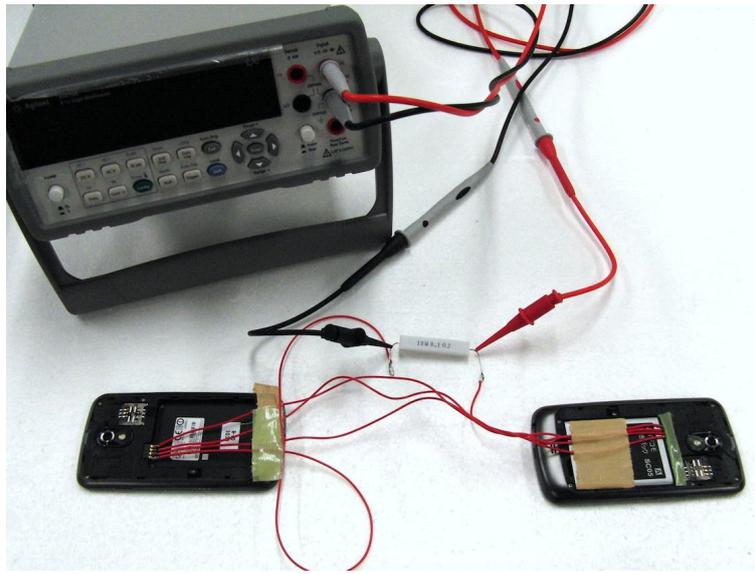


図 5.1: Android 端末の消費電力計測に用いた機器 (docomo GALAXY NEXUS SC-04D と Agilent 34410A デジタルマルチメータ)

たデータを逐次 MySQL データベースに保存していくように実装している [50] .

なお本研究において深く取り扱っていないが、細かい粒度で行動をモニタリングするためには各センサデータがどの時刻において取得されたものか同期をとる必要がある．前述したように、Android において加速度センサのサンプリング周波数を数値で指定することはできず、またアプリケーション中の処理によって周波数はリアルタイムに変動することがある．そのためなるべく正確な同期をとるためには、Network Time Protocol (NTP) などで端末の時計の時刻が同期されている上で、Android SDK において各サンプリングにおいてナノ秒の精度で与えられるタイムスタンプ *SensorEvent.timestamp* を利用することが望ましい．しかしながら、タイムスタンプの値から補間を行い、均等な時刻ごとのセンサデータを得てから圧縮を行う場合、補間に圧縮処理よりも多くの計算を必要とするため、端末の消費電力を増大させてしまう．本研究では簡単のためセンサデータの補間を行わず、各送信に含まれるデータをセンシングし始めた時刻のみを、圧縮された3軸加速度データに添えてサーバに送信することとした．

### 5.3.3 3G によるデータ通信後の IDLE 制御

ここで、図 5.2 に 3G による通信を行うときの通信デバイスの状態変化を示す．3G 通信を可能に設定している場合、端末は CELL\_PCH、もしくは IDLE と呼ばれる状態で待機している．データ送信を行うときの状態、ならびに送信後に基地局と専用のチャンネルを保持している状態は CELL\_DCH と呼ばれ、大きな電力を消費する．その後、連続通信時の遅延を削減するために基地局と端末の通信をいつでも再開できるよう低速なチャンネルを保持する CELL\_FACH と呼ばれる時間が存在する．このときの電力は CELL\_DCH 時に比べて小さいものではあるが、完全な IDLE 時に比べて無視できない電力を消費することが知られている [51]．これらの消費電力を低減するために、データ通信後速やかに IDLE 状態に移行する Fast Domancy と呼ばれるプロトコルや、他のアプリケーションとまとめて通信を行うことで状態変化の回数を低減させる方法などが提案されている．こう

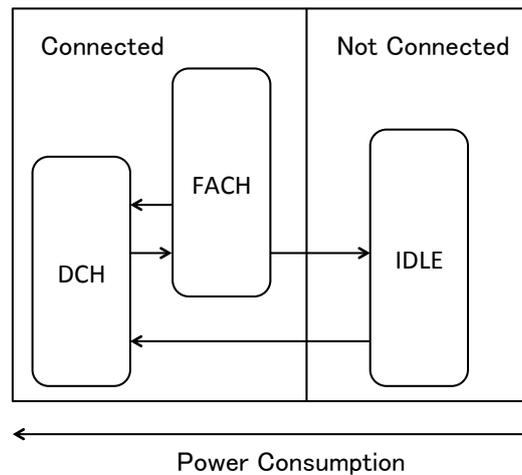


図 5.2: 3G によるデータ通信における通信デバイスの状態変化

した方法はしばしば基地局と端末間のシグナリングを増加させることでトラフィックに悪影響を及ぼすものであるが、1分に1回といったような頻繁に通信を行うアプリケーションにおいては、端末の消費電力を著しく低減させることが可能である。

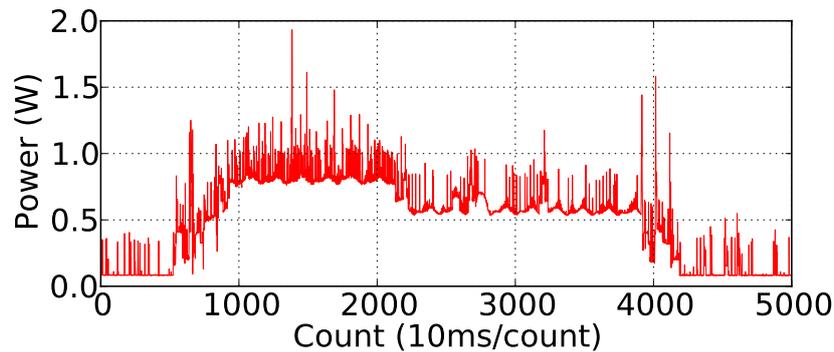
図 5.3(a) に、上述した機器を用いて実際に 3G によって 500kbyte のデータを送信したときの消費電力の時間変化を測定した例を示す。CELL\_FACH による時間は 20 秒持続しており、送信するデータ量によってはデータ通信そのものよりも CELL\_FACH 状態に電力を消費することが見てとれる。この状態は頻繁にセンサデータを送信する参加型センシングアプリケーションを想定した場合好ましい状態ではない。我々は各データ送信が終わった直後に通信のコネクションを強制的に切断することで、CELL\_FACH によって電力を消費する状況を回避し、なるべくデータサイズに比例する形で送信にかかる消費電力が増加するように実装と評価を行った。その実装を行った上で、500kbyte のデータを 3G を通じて送信したときの消費電力の時間変化を図 5.3(b) に示す。図 5.3(a) と比較して、CELL\_FACH による電力消費が無くなることで大幅にデータ送信にかかる電力が低減されていることが見てとれる。

## 5.4 各処理の消費電力測定結果

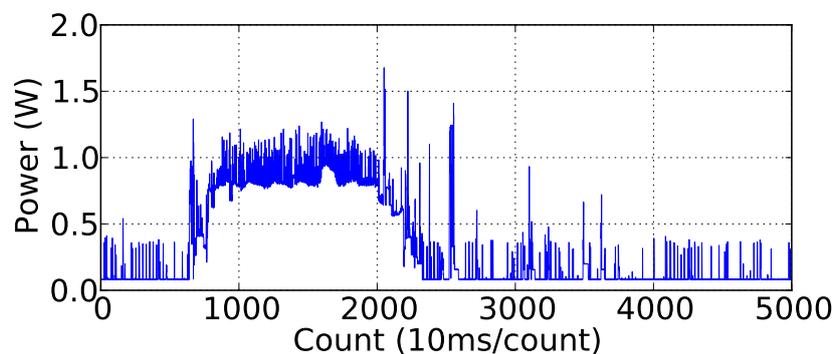
### 5.4.1 加速度センシング

図 5.4 に、GALAXY NEXUS において加速度センシングを行ったときの消費電力とサンプリング周波数を示す。提供されている 4 種類のサンプリングスピードを用いて ArrayList に加速度データを保存し続けるアプリケーションにおいて、サンプリング周波数はそれぞれ

- NORMAL: 15.5 Hz
- UI: 15.5 Hz
- GAME: 62 Hz
- FASTEST: 124 Hz



(a) 何の処理も行わなかった場合．20 秒の CELL\_FACH が存在する．



(b) データ送信後，即座にスリープに移行するよう実装した場合．

図 5.3: 3G によるデータ送信にかかる消費電力

となり，NORMALとUIにおいてはサンプリング周波数も消費電力も同じであったため，同じデータとして図 5.4 に示してある．また，加速度センシングを行わず，他に何の処理も行わないアプリケーションを起動しているときの消費電力を 0 Hz として図に示してある．サンプリング周波数が高ければ高いほど大きな電力を消費していることが分かる．また，加速度センサを駆動せず何の処理も行わないときでも，最も速いサンプリングスピードである 124 Hz でサンプリングを行うときの半分の電力を消費している．Prinyantha らが指摘しているように，加速度センサ自体の消費する電力は大きいものではなく，センシングに付随する CPU やメモリの処理によるものである [30]．センサを駆動していないときでも Android においては様々なバックグラウンド処理が働いており，CPU など他のコンポーネントは駆動している．そのためセンサ自体の駆動電力はそれらの消費電力に比べて大きな要素とはならず，低いサンプリング周波数のときは完全なアイドル時とそれほど消費電力が変わらない結果になっていると考えられる．

#### 5.4.2 データ圧縮

図 5.5 に，ZIP 圧縮もしくは圧縮センシングを用いたときの元のデータサイズとデータ圧縮にかかる消費電力量の関係を示す．ZIP 圧縮はハフマン符号化を含む比較的複雑な処理が必要となるため，圧縮センシングと比較して大きな電力を消費していることが分かる．圧縮センシングにおいては，圧縮処理は  $N$  次元のデータに対して  $M \times N$  次元の観測行列を掛けることによって行う．圧縮

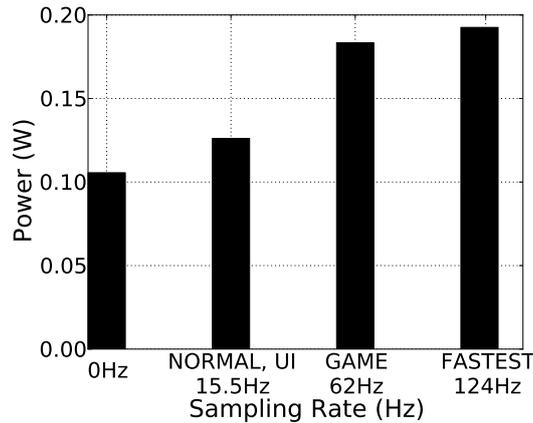


図 5.4: 加速度センシングの4つのサンプリング周波数とその消費電力

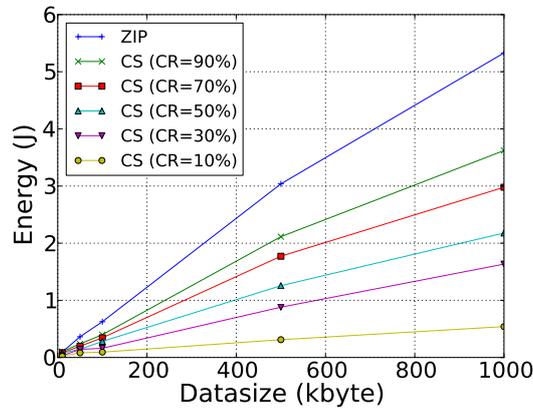


図 5.5: 圧縮方式ごとのデータ圧縮におけるデータサイズと消費電力量の関係

センシングの圧縮率は  $\frac{M}{N}$  によって定まるため、圧縮率が小さい、すなわち  $M$  の値が小さければ小さいほど、圧縮にかかる計算は少なくその分消費する電力も少ないものとなると考えられる。また、今回用いている観測行列は各行に4つの1を含みその他が0からなる疎行列であるため、 $N$ 次元のデータの圧縮にかかる計算は  $4 \times M$  回の足し算のみである。 $M$  の値が小さければ小さいほどデータ圧縮にかかる電力もデータ送信にかかる電力も小さくなるため好ましいが、それらは復元誤差とトレードオフの関係にある。同じ誤差を許した上で  $M$  の値を小さくするには、加速度データをよりスパースに表せる基底行列と観測行列の組み合わせを見つける必要がある。

### 5.4.3 データ送信

図 5.6 に、3G によるデータ送信におけるデータサイズと消費電力量の関係を示す。データサイズによって線形に消費電力量が増加することが分かる。ただ、CELL\_FACH による消費電力をカットするよう処理を行っても、データ送信自体にかかる消費電力以外のオーバーヘッドは大きい。ペイロードが 100 kbyte を超えないようなデータ送信では、データ送信そのものにかかる電力量よりも

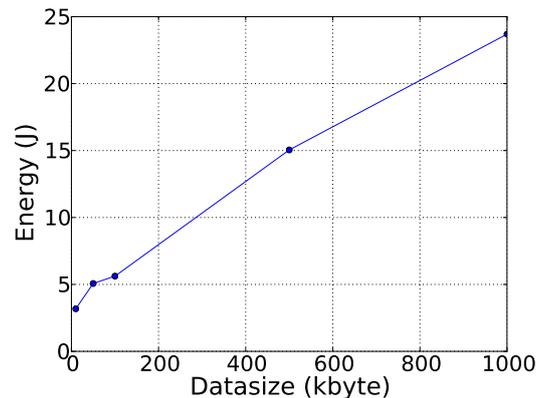


図 5.6: 3G によるデータ送信におけるデータサイズと消費電力量の関係

基地局とのシグナリングによって通信デバイスの状態を変化させることに電力を消費してしまうことが分かる。

## 5.5 全体の消費電力測定結果

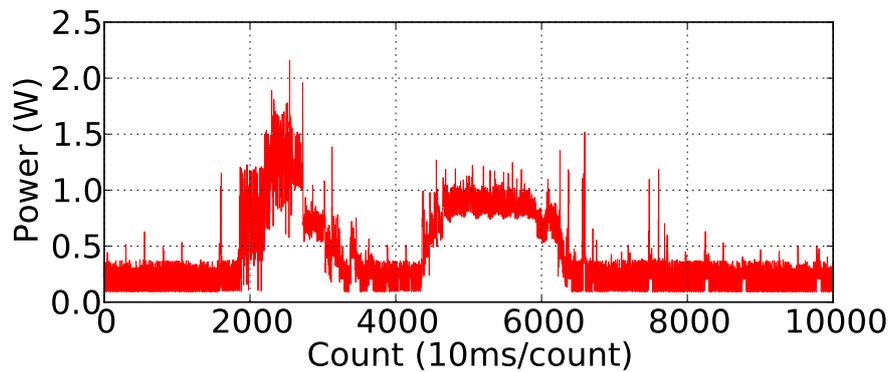
### 5.5.1 異なる圧縮方式における比較

最後に、これらの要素をあわせた全体の消費電力を評価する。データ圧縮とデータ送信の消費電力を比較しやすくするため、データ圧縮開始とデータ送信開始には5秒の時間差を設けて評価を行っている。図 5.7 に各圧縮方法を用いたときの圧縮とデータ送信にかかる消費電力の時間変化を示す。データ送信間隔は300秒であるが、この図は圧縮とデータ送信を行う100秒のみに注目して消費電力の時間変化を記録したものである。2つの図にある2カ所の消費電力の高い部分は、左側がデータ圧縮、右側がデータ送信によるものである。ZIP圧縮を行った場合に比べて、圧縮センシングを行った場合の方がCPUにかかる負担が少なく、消費電力が少なくなっている事が見てとれる。また、データ送信にかかる消費電力を同じにした上で圧縮にかかる消費電力を比較するため、圧縮センシングの圧縮率をZIPとほぼ同じである50%に設定しており、図 5.7(a) と図 5.7(b) の両方で、送信にかかる消費電力はほぼ同じとなっていることが分かる。

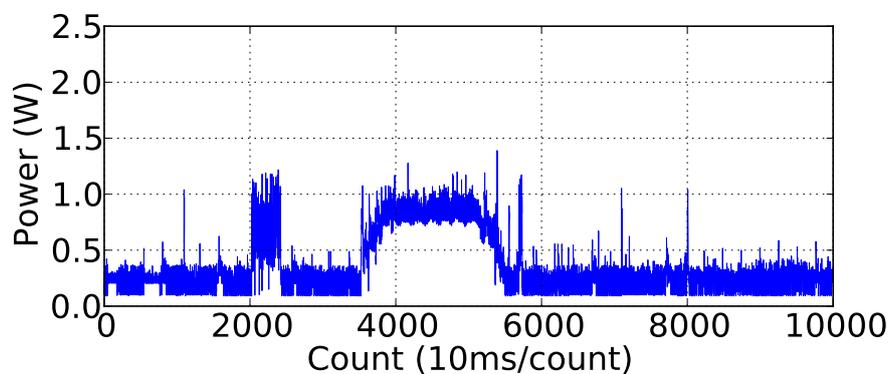
圧縮センシングを用いることで、ZIP圧縮を用いた場合と比較して、加速度センシングにかかる電力なども含めて全体で約6%消費電力を低減できた。これは圧縮センシングにおける圧縮アルゴリズムが軽量に行えることに由来したものである。

### 5.5.2 考察

GALAXY NEXUSにおいて、加速度センシングを行う、300秒経過した時点でデータ圧縮とデータ送信を行うというセンシングアプリケーションにおける処理の1周期において、消費される電力の内訳がどのようなになっているかに注目した。前節において評価した各要素ごとの消費電力を元に近似式を計算し、これらの処理の1周期における消費電力の内訳を算出した。図 5.8 に、圧縮を行わず生データを送信した場合、ZIP圧縮を用いた場合、圧縮センシング(圧縮率50%)を用いた場合のそれぞれの消費電力の内訳を示す。また、何の処理も行わず加速度センシングもしないアプリ



(a) ZIP 圧縮を行う場合



(b) 圧縮センシングを行う場合

図 5.7: ZIP 圧縮と圧縮センシングの圧縮とデータ送信にかかる消費電力の比較

ケーションを起動し、ディスプレイをオフにした状態における消費電力を Background Tasks による消費電力とした。

圧縮センシングを圧縮率 50% で用いた場合、生データを送る場合と比べて約 13%、ZIP 圧縮を用いる場合と比べて約 6% 消費電力を削減できる。また、データ圧縮とデータ送信にかかる消費電力のみで比較すると、生データを送る場合と比べて約 40%、ZIP 圧縮を用いる場合と比べて約 19% 消費電力を削減できる。

図 5.8 から、センシングアプリケーションが消費する電力の半分以上は加速度センシングによるものであることが分かる。センシング自体にかかる消費電力を下げる取り組みは既に多くなされており、主に GPS による位置取得の分野でデューティサイクルを設けるなどの方法がとられている。本研究はセンシングされたデータを省電力に遠隔サーバに収集する方法に焦点を当てたものであり、こうしたセンシングそのものに関する消費電力について対策を行っていない。しかしながら、本研究はセンサデータの圧縮センシングにより送信の電力コストを下げることをのみを行っているため、そうした既存のセンシングの省電力化方法と衝突せず、同時に使えるものである。そのためセンシングにかかる消費電力の低減技術と本研究による圧縮方法を組み合わせることで、更に消費電力が低減できると考えられる。

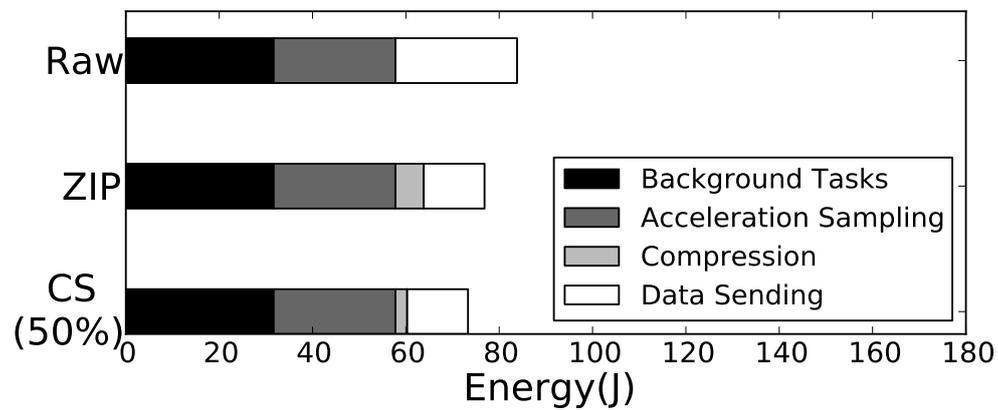


図 5.8: 圧縮方式の違いによる端末の消費電力の内訳の比較

## 5.6 評価のまとめ

観測行列に sparse random matrix を用い、基底行列に K-SVD を用いた圧縮センシング手法を実際の携帯端末における加速度センシングに適用したときの評価を行った。圧縮センシングを用いることで、ZIP 圧縮を用いてセンサデータを圧縮した場合に比べて全体で 6%、圧縮とデータ送信の部分だけで比較して 19% の電力を削減できることが分かった。消費電力の評価において圧縮率は 50% に定めており、そのとき復元による誤差は { 静止, 歩行, 走行, スキップ, 階段昇り, 階段降り } の 6 行動で平均して約 0.07 NRMSE に抑えることができる。これを行動認識手法に適用したとき、認識率は 75.5% であり、これは圧縮を行わない生データを用いて認識を行ったときと比べて、認識率の低下を 1.4% に抑えられている。

## 第6章

---

結論

## 6.1 本研究の主たる貢献

本研究では、携帯端末を用いた参加型加速度センシングにおいて、復元には複雑な計算を必要とするものの、圧縮処理が軽負荷に行える圧縮センシングを用いることで、圧縮処理に電力を多く消費することなく送信にかかるコストを削減し、携帯端末の消費電力を低減するシステムについて提案した。圧縮センシングは適用するデータのスパース性を利用して、ごく少ない観測数から精度高く元のデータを復元するアルゴリズムである。圧縮センシングに用いる観測行列としては、sparse random matrix を採用した。また基底行列には、機械学習を用いて訓練データをスパースに表現する行列を構築する手法である K-SVD を用いている。人間行動理解を目的としてボランティア的に集められた大規模な加速度データベースである HASC2010corpus を K-SVD の訓練データに用いることで、同じ行動をとっている加速度データ一般の圧縮センシングに利用可能な基底行列を構築した。圧縮センシングにおいて一般に用いられる DCT 行列と比較して、この K-SVD により構築した基底行列を用いることで復元による誤差を低減できることを示した。また、この圧縮センシング手法を用いて復元した加速度データに行動認識手法を適用したときに、ある程度まで圧縮率を下げても十分に行動認識が可能な認識精度を維持できることを示した。

提案手法の省電力性を評価するため、実際の Android 端末上にセンシングアプリケーションを実装し、デジタルマルチメータを用いて端末の消費電力を計測した。圧縮処理にかかる消費電力を明らかにし、典型的な情報圧縮である ZIP と比較して端末を省電力化できることを示した。

HASC corpus のような大規模加速度データベースを、K-SVD に加速度データベースを用いることで圧縮センシングに有効な基底行列を構築する試みは、本研究が初めてのものである。また、こうした圧縮センシング手法を適用して圧縮・復元を行った加速度データを行動認識に用いても認識が崩壊せず、十分に精度を維持できる事を示した点で貢献がある。

省電力性の評価については、スマートフォンにおいて圧縮センシングの圧縮処理にかかる消費電力と圧縮するデータサイズ、そして圧縮率の関係について明らかにした初めての取り組みである。実際の Android 端末上において、従来の圧縮方法よりも圧縮センシングの圧縮コストが非常に小さい事を評価し、データ送信やセンシングも含めて、実装したアプリケーション上でその省電力性を明らかにしたという点に貢献がある。

## 6.2 今後の課題

本研究において、以下の点が今後の課題として挙げられる。

### 6.2.1 観測・基底行列のサイズ最適化

本研究において K-SVD によって構築した基底行列のサイズは 200 行 1801 列としているが、最適な行列のサイズは定かではない。基底行列の列数を大きくしたとき一般に復元誤差は小さくなるが、復元処理にかかる計算量は増加する。行数を変化させた場合、つまり窓幅を変化させた場合にも復元処理にかかる計算量は増加するが、復元誤差がどのように変化するかは明らかではなく、また基底行列の種類にも依存する事象である。大量のデータを用いてあらゆる種類の基底行列について最適な窓幅を求めることは膨大な計算を必要とするが、復元誤差を最小化するためには必要な課題である。

### 6.2.2 行動ごとの圧縮最適化

図 4.4, 4.5 から分かることとして, 静止中の加速度データについては圧縮率をどんなに低く設定しても信号の概形は変化せず, 認識率も低下しない. 自宅から勤務地や学校に移動して仕事を行うといったルーティンワークを行う人々にとって, 1日にとる行動のうち静止している時間の占める割合は非常に大きい. そのため, 静止している時間は圧縮センシングの圧縮率を向上させる, もしくは加速度センシングを停止させるといった最適化を行うことで, 更なる省電力化が見込まれる.

### 6.2.3 実環境試験

第2章で述べた高齢者のヘルスマonitoringのための行動認識についてはアプリケーションの試作を行ったものの, それらを含めた一般的な参加型アプリケーションにおいて, 現実的な環境で実験協力者を得て実測試験を行ったわけではない. 本研究において用いた訓練データは HASC2010 corpus に含まれる 6 行動 { 静止, 歩行, 走行, スキップ, 階段昇り, 階段降り } の加速度データのみであり, 他の行動における復元誤差や認識率については明らかではない. また, 加速度データは個人によって値のばらつきが非常に大きく, 加速度データから行動認識を行う際には分類器を個人特化しないと著しく認識率が下がる傾向にある. こうした個人差や訓練データに含まれない他の行動への対応を検討するためには, 起床時から就寝時まで, また外出時間などにおける長時間にわたる実環境の加速度データを取得し, 分析を行うことが必要である.

## 謝辞

---

本研究および修士論文の執筆にあたり，親身にご指導くださり，また国内外における研究発表や学術訪問の機会を多く与えてくださいました浅見徹教授に，心より感謝致します．有り難うございました．

ならびに，研究の細部までご指導していただき，研究のみならず日常生活や就職活動においても常に適切な助言でサポートして下さった川原圭博講師に深く感謝致します．

研究を進めるにあたり，必要であった大規模な加速度データを提供してくださいました Human Activity Sensing Consortium (HASC) の方々，また，本研究による手法とウェアラブル血圧センサの併用に関しまして共同して研究を進めてくださった酒造正樹先生，Guillaume Lopez 先生にも謝意を表したいと思います．

また，工学部精密工学科から修士課程で情報理工学系研究科に進学し，大学院で新たな分野の学問を始める上で，分からない事ばかりであった自分を助けてくれるとともに，同じ立場として励ましあい，議論をしてくれた加藤拓也氏，佐々木達哉氏，繁田亮氏をはじめとする全ての同輩，先輩，後輩に感謝致します．

秘書の安藤さん，松岡さんをはじめ，日常の研究室生活を送る上でお世話になった浅見川原研究室の全ての皆様にお礼申し上げたいと思います．

最後に，研究以外の面において常に暖かいサポートで経済的，身体的，精神的に生活を支えてくれた家族に感謝したいと思います．

## 参考文献

---

- [1] N.D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury and A.T. Campbell, “A survey of mobile phone sensing,” *Communications Magazine, IEEE*, vol. 48, no. 9, pp. 140–150, September 2010.
- [2] S. Consolvo, D. W. McDonald, T. Toscos, M. Chen, J.E. Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, R. Libby, I. Smith and J. A. Landay, “Activity Sensing in the Wild: A Field Trial of UbiFit Garden,” *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pp.1797–1806, 2008.
- [3] E. Miluzzo, C. T. Cornelius, A. Ramaswamy, T. Choudhury, Z. Liu and A. T. Campbell, “Darwin Phones: The Evolution of Sensing and Inference on Mobile Phones,” *Proceeding of 8th International ACM Conference on Mobile Systems, Applications, and Services (MobiSys)*, San Francisco, CA, USA, pp. 5–20, June 2010.
- [4] The University of California Los Angeles: “Center for Embedded Networked Sensing”; <http://research.cens.ucla.edu/urban/>
- [5] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy and M.B. Srivastava, “Participatory Sensing,” *World Sensor Web Workshop, ACM SenSys 2006*, Boulder, Colorado, October 2006.
- [6] D. Estrin, “Participatory Sensing: Applications and Architecture,” *Proceeding of 8th ACM International Conference on Mobile Systems, Applications and Services (Mobisys)*, San Francisco, CA, pp. 3–4, Jun 2010.
- [7] NIKE+, <http://nikeplus.nike.com/plus/>
- [8] E. J. Candés and M. B. Wakin, “An Introduction To Compressive Sampling,” *IEEE, Signal Processing Magazine*, Vol. 25, no.2, pp.21–30, March 2008.
- [9] N. Eagle, A. Pentland and D. Laser, “Inferring Social Network Structure Using Mobile Phone Data,” *Proceedings of National Academy of Sciences (PNAS)*, Vol. 106, no. 36, pp. 15274–15278, 2009.
- [10] J. Cranshaw, E. Toch, J. Hong, A. Kittur and N. Sadeh, “Bridging the Gap between Physical Location and Online Social Networks,” *Proceedings of the 12th ACM International Conference on Ubiquitous Computing (UbiComp)*, pp. 119–128, Copenhagen, Denmark, September 2010.

- [11] D. Christin, A. Reinhardt, S.S. Kanhere and M. Hollick, “A survey on privacy in mobile participatory sensing applications,” *The Journal of Systems & Software*, vol. 84, no. 11, pp. 1928–1946, November 2011.
- [12] S. Reddy, A. Parker, J. Hyman, J. Burke, D. Estrin and M. Hansen, “Image browsing, processing, and clustering for participatory sensing: lessons from a DietSense prototype,” *Proceedings of the 4th Workshop on Embedded Networked Sensors (EmNets)*, pp. 13–17, 2007.
- [13] A. Madan, M. Cebrian, D. Lazer and A. Pentland, “Social Sensing for Epidemiological Behavior Change,” *Proceedings of the 12th ACM international conference on Ubiquitous computing (UbiComp)*, pp. 291–300, 2010.
- [14] E. Miluzzo, N.D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S.B. Eisenman, X. Zheng and A.T. Campbell, “Sensing meets mobile social networks: the design, implementation and evaluation of the CenceMe application,” *Proceedings of the 6th ACM conference on Embedded network sensor systems (SenSys)*, pp. 337–350, November 2008, Raleigh, North Carolina, USA.
- [15] S. Gaonkar, Jack. Li, R.R. Choudhury, L. Cox and A. Schmidt, “Micro-Blog: sharing and querying content through mobile phones and social participation,” *Proceedings of the 6th international conference on Mobile systems, applications, and services (MobiSys)*, pp. 174–186, 2008.
- [16] L. Deng and L. P. Cox, “LiveCompare: grocery bargain hunting through participatory sensing,” *Proceedings of the 10th ACM workshop on Mobile Computing Systems and Applications (HotMobile)*, pp. 1–6, 2009.
- [17] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G-S. Ahn and A. T. Campbell, “The BikeNet mobile sensing system for cyclist experience mapping,” *Proceedings of the 5th ACM international conference on Embedded networked sensor systems (SenSys)*, pp. 87–101, 2007.
- [18] E. Kanjo, “NoiseSPY: A Real-Time Mobile Phone Platform for Urban Noise Monitoring and Mapping,” *Mobile Networks and Applications*, Vol. 15, No. 4. pp. 562–574, August 2010.
- [19] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo and J. Eriksson, “VTrack: accurate, energy-aware road traffic delay estimation using mobile phones,” *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, New York, USA, pp.85–98, 2009.
- [20] 飯島 勝矢, 亀山 祐美, 秋下 雅弘, 大内 尉義, 柳元 伸太郎, 今井 靖, 矢作 直樹, ロペズ ギヨーム, 酒造 正樹, 山田 一郎, “高齢者におけるウェアラブル血圧センサーの臨床応用: ~ 認知機能およびストレス感受性からみた血圧短期変動評価への有用性の検討 ~ ,” *人工知能学会論文誌*, vol. 27, no. 2, pp. 40-46, 2012.

- [21] M. Labat, G. Lopez, M. Shuzo, I. Yamada, Y. Imai and S. Yanagimoto: “Wearable blood pressure monitoring system, Case study of multiplatform applications for medical use,” Proceedings of 4th International Conference on Health Informatics (HEALTHINF2011), pp. 156-163, Rome, Italy, January 2011.
- [22] G. Lopez, M. Shuzo, H. Ushida, K. Hidaka, S. Yanagimoto, Y. Imai, A. Kosaka, J. J. Delaunay, and I. Yamada: “Continuous blood pressure monitoring in daily life,” Journal of Advanced Mechanical Design, Systems, and Manufacturing, vol. 4, no. 1, pp. 179-186, 2010.
- [23] Weka. Available from: <http://www.cs.waikato.ac.nz/ml/weka/>
- [24] Weka for Android. Available from <https://github.com/rjmarsan/Weka-for-Android/>
- [25] T. Abdelzaher, Y. Anokwa, P. Boda, J. Burke, D. Estrin, L. Guibas, A. Kansal, S. Madden and J. Reich, “Mobiscopes for Human Spaces,” IEEE Pervasive Computing, vol. 6, iss. 2, pp. 20–29, 2007.
- [26] H. Lu, N. D. Lane, S. B. Eisenman and A. T. Campbell, “Bubble-sensing: Binding sensing tasks to the physical world,” Pervasive and Mobile Computing, vol. 6, iss. 1, pp. 58–71, 2010.
- [27] D. H. Kim, Y. Kim, D. Estrin and M. B. Srivastava, “SensLoc: sensing everyday places and paths using less energy,” Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys), pp. 43–56, 2010.
- [28] M. Musolesi, M. Piraccini, K. Fodor, A. Corradi and A. T. Campbell, “Supporting Energy-Efficient Uploading Strategies for Continuous Sensing Applications on Mobile Phones,” Proceedings of the 8th International Conference on Pervasive Computing (Pervasive’10). Helsinki, Finland, pp. 355–372, May 2010.
- [29] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari and N. Sadeh, “A framework of energy efficient mobile sensing for automatic user state recognition,” Proceedings of the 7th international conference on Mobile systems, applications, and services (MobiSys), pp. 179–192, June 2009.
- [30] B. Priyantha, D. Lymberopoulos and J. Liu: “LittleRock: Enabling Energy-Efficient Continuous Sensing on Mobile Phones,” IEEE Pervasive Computing vol. 10, no. 2, pp. 12–15, 2011.
- [31] R. Cristescu, B. Beferull-Lozano, M. Vetterli and R. Wattenhofer. “Network correlated data gathering with explicit communication: Np-completeness and algorithms,” IEEE/ACM Trans. on Networking, 14(1):41–54, Feb. 2006.
- [32] C. Luo, F. Wu, J. Sun and C. W. Chen. “Compressive Data Gathering for Large-Scale Wireless Sensor Networks,” ACM Mobicom 2009, 2009.
- [33] S. Yang and M. Gerla: “Energy-Efficient Accelerometer Data Transfer for Human Body Movement Studies,” IEEE SUTC, pp.304–311, California, USA, July 2010.

- [34] B.A. Olshausen and D.J. Field, "Sparse Coding of Sensory Inputs." *Current Opinion in Neurobiology*, vol. 14, no. 4, pp. 481–487, 2004.
- [35] 田中利幸, "圧縮センシングの数理," *Fundamentals Review*, vol. 4, no. 1, pp. 39–47, 2010.
- [36] Available from: <http://cvxr.com/>
- [37] R. Coifman, F. Geshwind and Y. Meyer, "Noiselets," *Applied and Computational Harmonic Analysis*, vol. 10, Issue 1, pp. 27–44, January 2001.
- [38] R. Berinde and P. Indyk, "Sparse recovery using sparse random matrices," MIT-CSAIL TR-2008-001, 2008, Technical Report.
- [39] E. J. Candés, Y. C. Eldar, D. Needell and P. Randall, "Compressed sensing with coherent and redundant dictionaries," *Applied and Computational Harmonic Analysis*, vol. 31, Issue 1, pp. 59–73, July 2011.
- [40] M. Aharon, M. Elad and A. Bruckstein, "The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representations," *IEEE Trans. Image Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.
- [41] A. Carroll and G. Heiser, "An Analysis of Power Consumption in a Smartphone," *Proceedings of the USENIX Annual Technical Conference*, pp. 1–12, Boston, MA, USA, June 2010.
- [42] N. Kawaguchi, N. Ogawa, Y. Iwasaki, K. Kaji, T. Terada, K. Murao, S. Inoue, Y. Kawahara, Y. Sumi and N. Nishio, "HASC Challenge: Gathering Large Scale Human Activity Corpus for the Real-World Activity Understandings," *Proceedings of the 2nd Augmented Human International Conference (AH 2011)*, Article No.: 27, Tokyo, Japan, March 2011.
- [43] Ron Rubinstein Softwares Available from: <http://www.cs.technion.ac.il/~ron-rubin/software.html>
- [44] L. Bao and S. S. Intille, "Activity Recognition from User-Annotated Acceleration Data," *Proceedings of 2nd International Conference on Pervasive Computing*, pp. 1–17, Vienna, Austria, April 2004.
- [45] J. Lester, T. Choudhury and G. Borriello, "A Practical Approach to Recognizing Physical Activities," In *Proceedings of 4th International Conference on Pervasive Computing*, pp. 1–16, Dublin, Ireland, May 2006.
- [46] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen and M. Srivastava, "Using Mobile Phones to Determine Transportation Modes," *ACM Transactions on Sensor Networks (TOSN)*, Vol. 6, No. 2, pp. 1–27, February 2010.
- [47] J. R. Kwapisz, G. M. Weiss and S. A. Moore, "Activity Recognition using Cell Phone Accelerometers," *ACM SIGKDD Explor. Newsl.* Vol. 12, No. 2, pp. 74–82, March 2011.
- [48] docomo NEXT series GALAXY NEXUS SC-04D サポート情報:  
<http://www.nttdocomo.co.jp/support/utilization/product/sc04d/spec.html>

- 
- [49] N. Thiagarajan, G. Aggarwal, A. Nicoara, D. Boneh and J. P. Singh, “Who killed my battery?: analyzing mobile browser energy consumption,” Proceedings of 21th international conference on World Wide Web (WWW '12), pp. 41–50, NY, USA, 2012.
- [50] web.py Available from: <http://webpy.org/>
- [51] N. Balasubramanian, A. Balasubramanian and A. Venkataramani, “Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications,” Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference (IMC '09), pp. 280–293, New York, NY, USA, 2009.
- [52] 3GPP Release 7: UE Fast Dormancy behavior, 2007. 3GPP discussion and decision notes R2-075251.

## ■ 発表文献

---

### 国際会議 / International Conferences

- [P1] D. Akimura, Y. Kawahara, T. Asami, “A Compressed Sensing Method for Human Activity Sensing using Mobile Phone Accelerometers, ”Proceedings of the 9th International conference on Networked Sensing Systems (INSS2012), pp. 1–4, Antwerp, Belgium, June 2012.
- [P2] D. Akimura, Y. Kawahara, T. Asami, “Reducing Power Consumption of Human Activity Sensing using Compressed Sensing, ” Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys 2011), pp. 355–356, Seattle, USA, November 2011. (poster)

### 研究会 / Domestic Workshops

- [P3] 明村大登, 酒造正樹, 川原圭博, “連続血圧モニタリングのためのスマートフォンへの行動認識手法の実装方法の検討,” 第12回人間情報学会講演会, September 2012. (poster) (学生部門 最優秀ポスター賞)
- [P4] 明村大登, 川原圭博, 浅見徹, “圧縮センシングを用いた低消費電力な行動認識手法の実装と評価,” 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 (DICOMO2012), 1E-2, pp. 148–156, July 2012. (優秀プレゼンテーション賞)

### 全国大会

- [P5] D. Akimura, Y. Kawahara, T. Asami, “Dictionary Optimization for Sparse Representation of Acceleration Data using K-SVD,” IEICE general conference, March 2013. (to appear)
- [P6] 明村大登, 川原圭博, 浅見徹, “圧縮センシングを用いた行動センシング手法における復元誤差の認識精度への影響評価,” 電子情報通信学会 総合大会, B-15-7, March 2012.
- [P7] 明村大登, 川原圭博, 浅見徹, “行動センシングのための圧縮センシングによるデータ転送量削減手法,” 電子情報通信学会 ソサイエティ大会, B-19-7, September 2011.

# 付録

## K-SVD のアルゴリズム

K-SVD のアルゴリズムの簡単な流れを示す .

### 目的

$$\min_{D, X} \{\|Y - DX\|_F^2\} \quad \text{subject to} \quad \forall i, \|\mathbf{x}_i\|_0 \leq T_0 \quad (6.1)$$

を解いて、訓練データ  $Y$  をスパースな係数行列  $X \in \mathbb{R}^{K \times N}$  として表す基底行列  $D \in \mathbb{R}^{n \times K}$  を求める .

### 入力

$N$  個の  $n$  次元訓練データ  $(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N) \in \mathbb{R}^{n \times N}$  を各列に持つ行列  $Y$  を入力する .

### 1. 初期化

$Y$  からランダムな  $K$  列を抜き出し、各列を正規化して  $D$  とする .

### 2. 更新

以下の処理を  $T$  回繰り返す .

- $i = 1, 2, \dots, N$  において以下の式を解いて  $X$  の各列を求める .

$$\min_{\mathbf{x}_i} \{\|\mathbf{y}_i - D\mathbf{x}_i\|_2^2\} \quad \text{subject to} \quad \|\mathbf{x}_i\|_0 \leq T_0 \quad (6.2)$$

- $k = 1, 2, \dots, N$  において以下の処理を行って  $D$  と  $X$  を更新する .

-  $\omega_k = \{i \mid 1 \leq i \leq K, \quad \mathbf{x}_T^k(i) \neq 0\}$  を定義する .

-  $E_k = Y - \sum_{j \neq k} \mathbf{d}_j \mathbf{x}_T^j$  を計算する .

-  $E_k$  のうち  $\omega_k$  に含まれる番号の列のみを抽出した行列を  $E_k^R$  とする . 同じく、 $\mathbf{x}_T^k$  のうち  $\omega_k$  に含まれる番号の列のみを抽出した列、すなわち零でない成分を持つ列のみを抽出して  $\mathbf{x}_R^k$  とする .

- 特異値分解を行い、 $E_k^R = U \Delta V^T$  とする .

- $d_k$  に  $U$  の 1 列目を代入する .
- $x_R^k$  に  $V$  の 1 列目に  $\Delta$  の 1 行 1 列目の成分を掛けたベクトルを代入する .