Vol.2016-MBL-78 No.12
Vol.2016-UBI-49 No.12
2016/2/29

圧縮センシングを用いたセンサアプリケーションの開発効率化を目的としたシステム構成法

谷川 博哉[1]　満田 成紀[2]　福安 直樹[2]　松延 拓生[2]　鯵坂 恒夫[2]

概要：近年，モバイル端末やウェアラブルデバイスの普及が進み，多様なセンサから得られるデータを利用したアプリケーションの開発が盛んに行われている．しかし，センサから得られる多量のデータを効率良く処理するには，端末やデバイスの特性を考慮して，適切なシステムの設計や実装を行う必要がある．そこで本研究では圧縮を軽負荷で行うことができる圧縮センシング技術を，ウェアラブルデバイスを用いたセンサアプリケーションに組み入れる手法について検討する．アプリケーションの種類や，実データを対象としたシミュレーションの結果に基づき，センサアプリケーションの開発を効率化するシステムの構成法を提案する．

1. はじめに

スマートフォンやタブレット端末などのモバイル端末が普及し，センサから得られるデータを利用したアプリケーションが多数利用されている．また AndroidWear や AppleWatch などのウェアラブルデバイスも登場し，モバイル端末と組み合わせることで多彩なセンサから情報を取得することも可能となった．これによりセンサアプリケーションは今後もよりいっそう発展していくと考えられる．しかし，モバイル端末の発展に伴い高機能化されているものの，いまだ計算資源や電源には制限がある．この問題はウェアラブルデバイスでも取り上げられ，デバイスの設計レベルで議論がなされている．

センサから得られるデータは多く，モバイル端末で動作するセンサプリケーションでは効率的な処理が必要となる．センサノードとしての役割をウェアラブルデバイスが担い，親機であるモバイル端末を介してアプリケーションサーバまでデータを送信するシステム構成 (図 1) を想定した場合，モバイルアプリケーションにおける電力消費の要因の一つとしてデータ通信があげられる．センサから得られたデータの特徴量をウェアラブルデバイスで抽出し，観測したい部分のみを送信することによる通信効率化や，OS の処理系統に注目し不要な通信を削減することによる効率化など，デバイスや，OS レベルでの研究が行われている [1][2].

[1] 和歌山大学大学院システム工学研究科
Wakayama Univ., Graduate School of System Engineering
[2] 和歌山大学大学院システム工学部
Wakayama Univ., Faculty of Systems Engineering

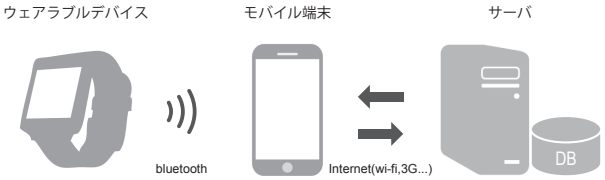

図 1 ウェアラブルデバイスを用いたアプリケーションのシステム構成例

アプリケーション上で，センサデータを捨てること無く通信コストを抑えるためにはデータの圧縮を行うことが有効であるが，圧縮の計算には多大なコストが発生する．そこで，軽負荷な計算で圧縮処理を行える圧縮センシングに注目し，ウェアラブルデバイスを用いたシステム構成に適用することで圧縮と通信コストの効率化を行うことができる．圧縮センシングを用いたモバイル端末の省電力化先行研究において効率化が確かめられている．本研究では，圧縮センシングをウェアラブルデバイスを用いたセンシングアプリケーションに組み入れる手法について検討する．

2. 圧縮センシングを用いた省電力化

2.1 圧縮センシング

圧縮センシングは，観測対象が多くの零成分と少数の非零成分で構成されるスパース性を持つと仮定した時に，高次元の信号を少数の観測データから復元する技術である．圧縮センシングは標本化定理で示されるサンプル数よりも少ないサンプルで観測対象の再構成を行える [3].

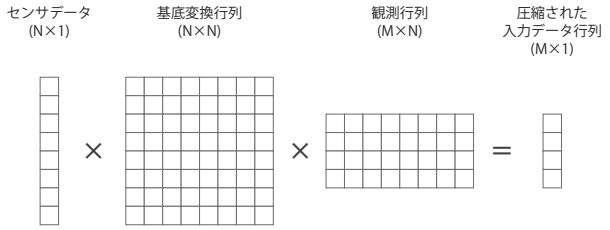

図 2　圧縮センシングにおける圧縮

2.2　圧縮センシング

実在する多くの信号を何らかの基底上で表現したときに，大部分の要素が 0 に近い形となり信号の持つ情報が一部の要素に集中する．これを圧縮可能性のある信号という．本稿では圧縮可能性のある信号についてもスパース性を持つ信号とする．元の信号がスパース性を持つとき，圧縮後の信号から L1 ノルム最小化問題を解くことで元信号を推定することができる．

2.2.1　圧縮アルゴリズム

観測対象を N 次元の信号 $x \in \mathbb{R}^N$ であるとする．$M \times N$ の観測行列 $\Phi \in \mathbb{R}^{M \times N}$ を用いて式 (1) のように線形変換されているものとする．

$$y = \Phi x \in \mathbb{R}^M \tag{1}$$

上記の式 (1) にセンサデータを当てはめることで圧縮センシングにおける圧縮を行う (図 2)．圧縮処理は観測行列を元信号に乗算することで行われるため，計算量の少ない観測行列を選択することで，複雑な計算を必要とせず圧縮にかかる計算コストを削減することができる [4]．

2.2.2　復元アルゴリズム

信号 $x \in \mathbb{R}^n$ がスパース性を持つ時，N 次元の信号 $s \in \mathbb{R}^N$，$N \times N$ の基底行列 $\Psi \in \mathbb{R}^{N \times N}$ を用いて信号 x をスパースに表現できるとする．

$$x = \Psi s \in \mathbb{R}^N \tag{2}$$

s の非零成分が未知の場合，式 $y = \Phi x = \Phi\Psi s$ かつ，スパースな s を推定する問題として圧縮センシングの復元を表す事ができる [5]．

2.3　圧縮センシングを用いた省電力化

圧縮センシングの技術は，2.1 節の圧縮アルゴリズムを用いることで，既存の情報圧縮技術と比較して少ない計算量で圧縮を行うことができる．そのため，計算や通信による電力消費を低減することが可能である．圧縮センシングの技術は無線センサーネットワークなどを対象として研究されている [6]．特に，長時間に渡ってセンシングの必要があるセンサの電力消費を削減することが可能である [7]．モバイル端末においても同様に圧縮を行うことで端末の電力消費を抑えることが可能である [5]．

ウェアラブルデバイスをセンサノードとして扱うことで圧縮センシングの技術を応用できるのではないかと考えられる．そこで本稿では，ウェアラブルデバイスを用いたアプリケーションに注目し，ウェアラブルデバイス上で圧縮を行うことで，ウェアラブルデバイスおよびモバイル端末の処理の効率化を図る．

3.　圧縮の前処理における基底変換シミュレーション

圧縮センシングでは圧縮前の元信号のスパース性を高めることが復元の精度を高めることにつながる．そこで，ウェアラブルセンサアプリケーションのシステム構成を検討するために，ウェアラブルデバイスから得られたセンサデータの基底変換を行った際のスパース性を検証する．実際の端末から取得されたセンサデータから離散コサイン変換（DCT）を行った行列と変化の差分を取った行列を求め，スパース性について比較を行った．まず，ウェアラブルデバイスを手首に取り付けた状態で日常動作を行い，シミュレーションに用いるためのデータを取得する．そして，取得したデータを用いて基底変換のシミュレーションを行い，圧縮の前処理においてスパース性の変化を比較する．

3.1　実験概要

実際の Android 端末とウェアラブルデバイスにセンサデータを取得・送信・保存するアプリケーションを実装し，センサデータを取得する．取得されたセンサデータを用いて，基底変換を行い変換後の信号のスパース性を確認する．

センサデータの取得には LG G Watch R を用いた．今回取得したセンサの種類は LG G Watch R に搭載されたセンサの内の加速度センサ，心拍センサである．加速度データは 6300 個の時系列データを取得した．取得されたデータはサンプリング周波数 30Hz で 5 分間，日常動作（座る・立つ・歩く・物を持つなど）を行ったものを測定した (図 3)．加速度データは行動認識などに用いられる事を想定してユークリッドノルムを計算し重力加速度 9.8 で減算した形で用いた．心拍データについても同様に 1800 個の時系列データを取得した．取得されたデータはサンプリング周波数 1Hz で 30 分間測定した (図 4)．

取得したデータをスパース行列に変換する基底として，離散コサイン変換 (DCT) と前後のデータの差分を取った行列への変換を利用する．以降，離散コサイン変換を用いて基底変換を行ったものを DCT 基底，隣接要素との差分を取ったものを差分基底と表す．取得したセンサデータをアプリケーションで使用することを考え，それぞれに DCT 基底，差分基底による変換を行った．加速度センサのデータは，動きの大きい部分と小さい部分の 2 種類について 100 件，300 件，1800 件を抜き出したものに変換を行った．また，5400 件のデータを抜き出したものに対しても変換を

Vol.2016-MBL-78 No.12
Vol.2016-UBI-49 No.12
2016/2/29

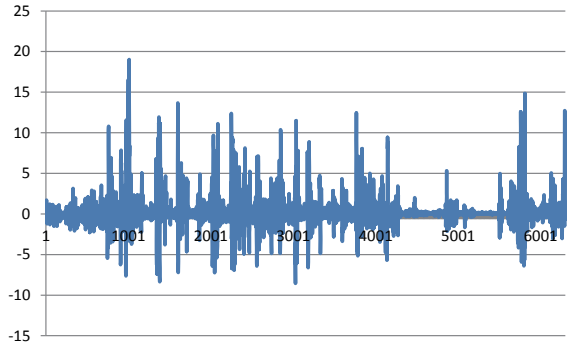

図 3　取得した加速度データ

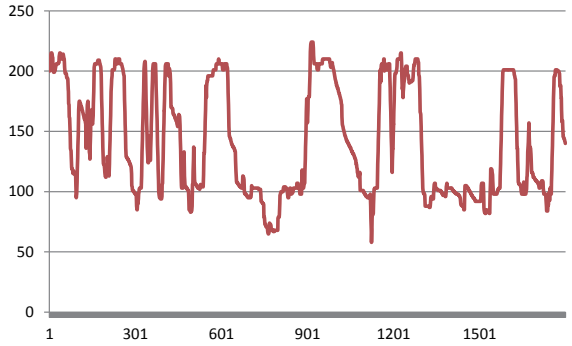

図 4　取得した心拍データ

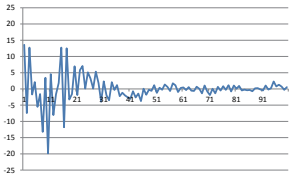

図 5　DCT 基底変換した
動きの大きいデータ

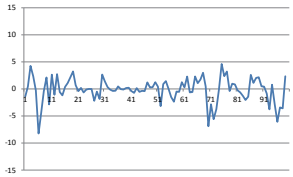

図 6　差分基底基底変換した
動きの大きいデータ

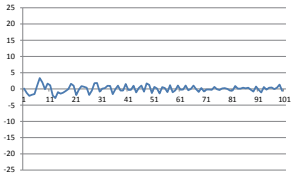

図 7　DCT 基底基底変換した
動きの小さいデータ

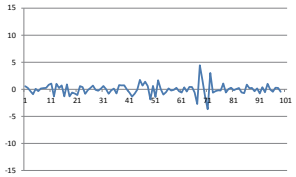

図 8　差分基底基底変換した
動きの小さいデータ

行った．心拍データは，15 件，50 件，300 件，1500 件のデータを抜き出して変換を行った．

3.2　実験結果

3.2.1　加速度センサ

動きの大きい部分の加速度データ 100 件に対して DCT 基底変換を行ったものが図 5，差分基底変換を行ったものが図 6 である．

DCT 基底による変換を行った場合は一部の信号に情報が偏っていることが確認できる．差分基底では，零成分が

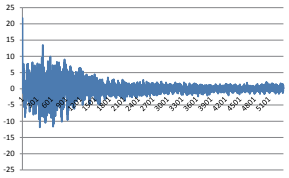

図 9　DCT 基底変換した
5 分間の加速度データ

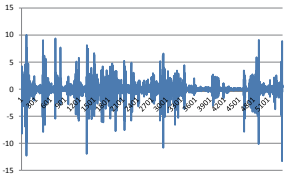

図 10　差分変換した
5 分間の加速度データ

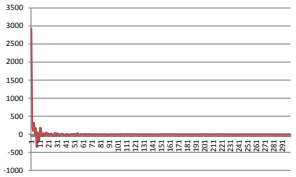

図 11　DCT 基底

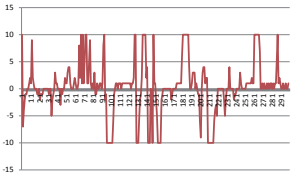

図 12　差分基底

少なくスパースであるとは言いがたい．動きの小さい部分の加速度データ 100 件に対して DCT 基底変換を行ったものが図 7，差分基底変換を行ったものが図 8 である．DCT 基底による変換と差分基底による変換共に，零成分が多くスパースではあるが同時に非零要素も少ない結果となった．3 秒間，10 秒間，1 分間とデータ量を増やす毎に動きの大きな部分のデータと小さな部分のデータが混在するようになるため，DCT 基底変換での情報量は全体に均一化されていき，差分基底では零成分の要素が増えていく．5 分間分のデータについても同様の結果が得られた．(図 9，10)

3.2.2　心拍センサ

抜き出した 300 件のデータに DCT 基底変換を行ったものが図 11，差分基底変換を行ったものが図 12 である．心拍センサは滑らかな波形を描くことが図から読み取れる．滑らかな波形を描くデータは DCT 基底においてスパースな信号へ変換することができる．差分基底と DCT 基底を比較すると，明らかに DCT 基底の方が非零成分が少なく，スパースであると言える．15 件，50 件，1500 件についても同様に DCT 基底においてスパースな信号へと変換することができた．

4.　圧縮センシングを用いたウェアラブルアプリケーションのシステム構成の提案

圧縮の前処理をシミュレーションした結果，センサの種別・取得頻度などを元に適切な基底を選択することがスパース性を高められると確認できた．圧縮の前処理として行列の変換を行う際に，複数の手法を切り替えて利用することでより復元精度を高めることができると考えられる．実験で得られた結果を元に，ウェアラブルデバイスを用いたセンサアプリケーションの開発において通信の効率化を行うシステム構成を提案する．本章では，提案するシステム構成の詳細及び，アプリケーション，センサの種別への対応方法について述べる．

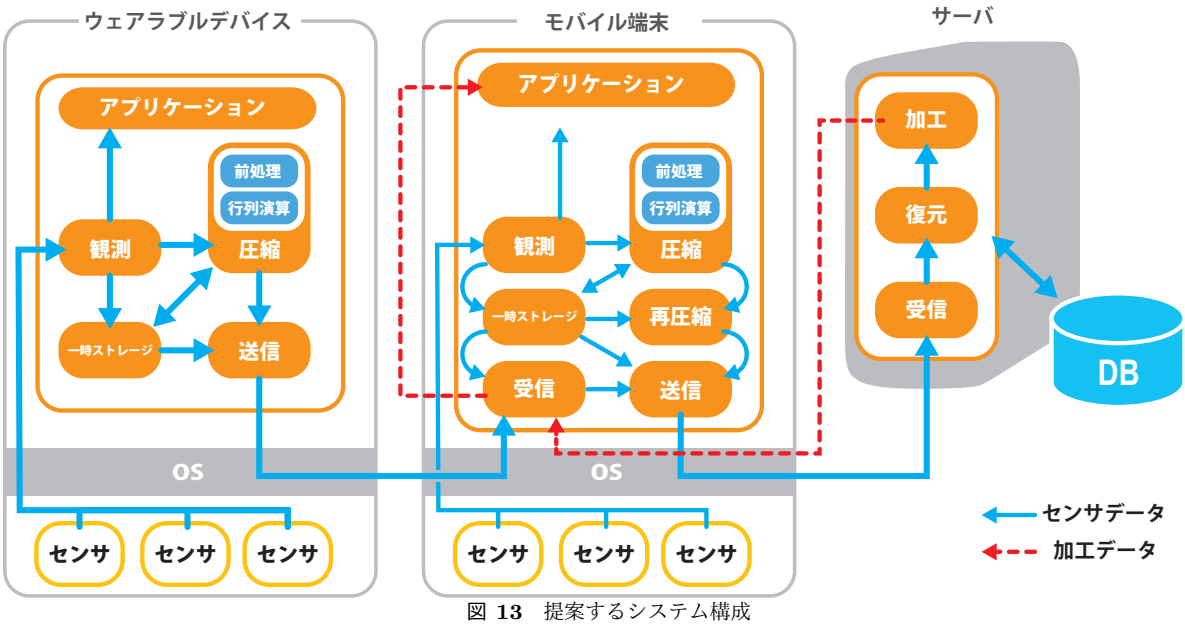

図 13　提案するシステム構成

4.1　構成

本稿で提案するシステム構成を図 13 に示す．提案するシステム構成では，様々なセンサアプリケーションへの応用を想定し，それぞれの機能を自由に組み合わせる設定を行えるように設計する．得られたデータの圧縮にはスパースランダムマトリクスによって生成された観測行列を用いる．

4.1.1　ウェアラブルデバイス

ウェアラブルデバイスはセンサデータの観測部，圧縮部，送信部，一時ストレージ機能を持つ．

観測部

設定に応じてセンサへの問合せを行う．観測の頻度や複数のセンサへの問合せを一括して設定できる機能を持つ．アプリケーションの設計に合わせて，アプリケーションに観測したデータを返す．圧縮の有無を切り替える．

圧縮部

センサ毎に取得したデータに対して前処理を行った後，観測行列を掛けあわせて圧縮を行う．前処理はセンサの種別・取得頻度・データの使用方法などに合わせて基底を切り替える機能を持つ．また，観測行列の次元を設定することで，アプリケーションに合わせた圧縮率・圧縮間隔の設定を行える．圧縮頻度によって，圧縮後のデータを通信部に渡すか，一時ストレージに格納するかを選択できる．

送信部

圧縮したデータをモバイル端末へ送信する．複数のセンサを利用する場合に合わせて，データの送信間隔，圧縮間隔を調整する機能を持つ．複数の圧縮されたデータをモバイル端末へ送信する場合は，一時ストレージへ問合わせる．

一時ストレージ

圧縮の頻度，送信の間隔に合わせて一時的にデータを格納する．圧縮前のデータを格納する機能と圧縮後の機能を格納する機能を持つ．

4.1.2　モバイル端末

モバイル端末ははセンサデータの観測部，圧縮部，送信部，一時ストレージ機能に加えて，データの受信部，再圧縮部を持つ．

観測部

モバイル端末に搭載されたセンサへ問合せを行う．ウェアラブルデバイスの観測部と同様の機能を持つ．

圧縮部

ウェアラブルデバイスの観測部と同様にデータの圧縮を行う．モバイル端末に搭載されたセンサに合わせて前処理の設定を行うことができる．

送信部

サーバへ圧縮されたデータを送信する．ウェアラブルデバイスにおける送信部と同様の機能を持つ．圧縮されたデータを一括送信する際に，再圧縮を行うかを設定することができる．再圧縮を行う場合は再圧縮部に問い合わせる．またサーバとの通信が確認できない場合は一時ストレージへデータを渡す．

一時ストレージ

観測部から送られてきた圧縮前のデータを格納する機能と圧縮部から送られてきた圧縮後の機能を格納する機能を持つ．加えて，ウェアラブルデバイスから送信されてきたデータを一時的に格納する．

受信部

ウェアラブルデバイスから送信されてきた圧縮されたデータを，送信間隔の設定にしたがって送信部，または一時ストレージに渡す機能を持つ．また，アプリケーションの設計に合わせて，ウェアラブルデバイスから送られてきた未圧縮のデータを圧縮部へ渡す，サーバ

にから送られてきたデータをアプリケーションに渡すなどの仲介を行う．

再圧縮部

サーバへのデータ送信間隔や，一度の送信におけるデータ量に合わせて既存の圧縮手法を用いて圧縮済みのデータの再圧縮を行う．再圧縮するデータの種類に合わせて圧縮手法を選択できる．

4.1.3 センササーバ

サーバでは，データ受信部，復元部，データベース，データの加工部を持つ．

受信部

モバイル端末から送信されてきた圧縮されたデータを受信する．復元部へデータを渡す．

復元部

圧縮されたデータを復元し，データベースに保存する．再圧縮が行われたデータの場合には，解凍後，復元を行う．復元にはセンサ毎に設定された観測行列と，圧縮時に設定された基底を用いる．復元したデータはデータベースに登録される．

データベース

復元されたセンサデータを蓄積する．

加工部

アプリケーションの設計に合わせて，復元されたデータを加工しアプリケーションへ送信する．

4.2 センサ種別への対応

本提案では，利用するセンサ毎に設定を行い，機能を切り替えることでセンサの種別に対応する．

4.2.1 取得できる値に対する対応

利用するセンサの種別が変われば，取得できる値やその変動の激しさが変わる．圧縮部では，センサの種別ごとに基底を設定する．加速度センサやジャイロセンサなどの数値の変化が激しいセンサに対しては，取得頻度・送信間隔をアプリケーションの目標に合わせることができる．心拍センサや気圧センサのような数値の変動が滑らかなセンサを扱う際は，一時ストレージを活用することで通信回数の削減を行う．

4.2.2 取得間隔に合わせた対応

センサの取得間隔の違いによって，送信するデータの大きさや，圧縮・送信を行うタイミングが変わってくる．取得間隔が短い場合，圧縮部で圧縮頻度を調整し，圧縮後一時ストレージに置いておき，一括で送信することで通信コストを抑えることができる．一度に圧縮するデータの数が多くなった場合は，圧縮部で計算量の少ない前処理を選択することや，モバイル端末上で再圧縮することで対応する．また，一度に圧縮するデータ量を減らすために，送信間隔と圧縮間隔を別に設定しておくことも可能である．取得間隔の長いセンサに対しては，圧縮前のデータを一時ストレージに格納することで圧縮・送信の回数を減らすことができる．

4.3 アプリケーションの目的に合わせた対応

アプリケーションの目的によって，センシングを継続する時間の調整や，圧縮する前のデータの利用が必要になる可能性がある．本提案では，ウェアラブルデバイス，モバイル端末，サーバのそれぞれに与える役割を切り替えることで対応する．長時間連続してセンシングするアプリケーションの場合，ウェアラブルデバイス上で圧縮・送信回数を調整し，またモバイル端末上でも圧縮されたデータを再圧縮することでサーバへの通信を減らす事ができる．ウェアラブルデバイスで得られたデータをモバイル端末上のアプリケーションで利用する場合，圧縮されたデータをモバイル端末に渡してしまうと，端末上で復元する必要が生じる．これに対応するために，ウェアラブルデバイスで圧縮を行わず，モバイル端末上で圧縮を行えるように機能を切り替えることで対応する．

5. 考察

3 節で行ったシミュレーションと，4 節で提案したシステム構成について考察を行う．

5.1 センサの前処理について

本稿では実際のウェアラブルデバイスから得られたデータに対して基底変換のシミュレーションを行った．実験では加速度データについて基底変換を行った際に，ランダムな動きが続く部分では DCT 基底・差分基底共に非零成分が多く出てしまうという結果になった．また，本稿で基底として採用した DCT 基底と差分基底以外に提唱されている K-SVD などを基底として取り入れることで，復元率を低下させることなく圧縮率を高めることができる可能性がある [8]．

5.2 センサの種類について

3 節で行ったシミュレーションに用いたセンサ 2 種について，取得頻度とセンサデータの値について比較を行う．加速度センサは取得頻度が高く，値の変動が大きい．心拍センサは取得頻度が低く，値の変動は小さい．そこから，実験で確認出来なかったセンサの種類として，
取得頻度が高く，値の変動が小さい
取得頻度が低く，値の変動が大きい
が推測される．これらのセンサに DCT 基底，差分基底を適用した場合について検証する必要がある．

5.3 実際のシステム運用について

実際にシステムを運用するとセンサデータの取得失敗や，サーバとのアクセスが途切れるなどの障害が発生する．提

案手法のシステム構成では，サーバとネットワークが切断された場合にアプリケーション上では圧縮されたデータを復元できない．データの復元を行うためには，多大な計算コストが発生する事になり，本研究の目的であるコスト削減にはつながらない．

あらかじめアプリケーションで利用するデータの圧縮サンプルをモバイル端末に与えることで，サンプルと圧縮されたデータの一致度を見れば圧縮前のデータを推測する事ができる [9]．これにより一定のセンサデータの変化を検知するアプリケーションなどにおいては，スタンドアロンな状態においても提案したシステム構成で対応することができる．

複数の提案手法を用いたアプリケーションを同時に利用する状況も考えることができる．複数のアプリケーションに対応するためには本提案手法のサービス化を行うことで，複数のアプリケーションにまたがってセンサデータを一括で圧縮送信することが可能となる．

6. まとめ

本研究では，圧縮センシングの技術の圧縮部分に着目し，実際のウェアラブルデバイスから得られたデータに適用してシミュレーションを行った．結果から得られた情報を元に設計を行い，ウェアラブルデバイスを用いたアプリケーションを効率化するシステム構成を提案した．

提案手法を用いることで，複数のセンサを 1 つの仕組みで扱うことができ，アプリケーションの設計・開発の手間を省くことができる．提案手法ではアプリケーションの設計の自由度を維持するために，センサの扱い方はアプリケーション依存であり，開発時にユーザが手動で設定する必要がある．設定の自動化ができればさらなる効率化を見込むことができる．

参考文献

[1] 桑原啓, 樋口雄一, 小泉弘, 笠原亮一：スマホで視る血液の流れー超小型　ウェアラブル血流センサ, NTT ジャーナル Vol.26 No.11,pp.21-24,2014
[2] 早川愛, 磯村美遊, 竹森敬祐, 山口実靖, 小口正人：Android 端末省電力化のためのブロードキャストインテント情報の調査, マルチメディア, 分散, 協調とモバイル (DICOMO2014) シンポジウム,2014
[3] 三村 和史: 圧縮センシング-疎情報の再構成とそのアルゴリズム- , 数理解析研究所講究録, 第 1803 巻,pp26-56,2012.
[4] E.J. Cand ěs and MB. Wakin: An introductien to cempressive sampling, IEEE Signal Processing Magazine,vol.25,No.2,pp.21-30,2008.
[5] 明村大登: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, 東京大学大学院情報理工学研究科電子情報学専攻 修士論文,2013.
[6] 黒岩 拓人, 鈴木 誠, 猿渡 俊介, 長山 智則, 森川 博之：無線センサネットワークを用いた構造モニタリングのためのマルチチャネル利用型並列一括収集機構, 電子情報通信学会論文誌, Vol.J96-B, No.2, pp.114-123,2013.
[7] 河上 春樹，川原 圭介，木代 雅巳, 工藤 高裕, 浅見 徹: 圧縮センシングに寄る高速道路加速度センサの消費電力削減, 電子情報通信学会ソサイエティ大会講演論文集,2013 年_通信 (2),pp. 431,2013.
[8] Michal Aharon, Michael Elad, and Alfred Bruckstein: K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation, IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 54, NO. 11,pp.4311-4322, 2006.
[9] 岩崎 正裕，藤波 香織：圧縮センシングを用いた指差し呼称時の腕振り動作認識に関する研究，電子情報通信学会技術研究報告. PRMU, パターン認識・メディア理解 112(441),pp.121-122,2013.