

節電の取り組みを可視化する Android アプリの実装と評価

大久保成晃¹ 菊池浩明² 石井啓之¹

概要: 本稿では, 公共施設での節電の取り組みの結果を, 個人として確認ができない課題を研究する. ひとりでエレベータの利用を控えて階段を進んで使ったとしても, 個人での取り組みが形に残らず, 節電に貢献している実感が薄い. 家庭では, 節電の取り組みは最終的に電気代の請求書という形で本人に返り, どれだけ節電しているか実感がわくものに対して, 階段の利用は証拠が残らない. そこで, 本研究では公共の場における節電のみえる化を行い, 個人がどの程度の貢献ができたのかを可視化する. 日常の動作から個人の状態を推定し, 節電量を可視化することで, 家庭と同様に利益が発生する仕組みを考える. 階段やエレベータという状態を, Android 端末に掲載されるセンサを用いて, 利用者の状態を自動推定する Android アプリケーションを開発する. 本システムには次の特徴がある, (1) ユーザの四状態 (歩行, 階段, エレベータ, 停止) を自動判定する, (2) ユーザが階段を使って節電した量を電力に換算して表示する.

Design and Implementation of Android App Visualizing Power Saving

NARIAKI OHKUBO¹ HIROAKI KIKUCHI² HIROSHI ISHII¹

1. はじめに

東日本大震災による原発の事故により電力供給量の低下とそれに伴う脱原発の動きがあり, 節電の必要性が高まってきた. 企業や官公庁, 学校など多くの組織が様々な方法で節電に取り組んでいる. 例えば, 冷房の設定温度を上げたり, クールビズ, サマータイムの導入, エレベータの稼働数を減らす「上下3階以上の移動には階段を使いましょう」といった取り組みやパナソニックの節電取り組み [1] や, [2] などがある. 更に, 個人での取り組みにまで要請されている. 例えば, スマート家電による自宅の家電の消し忘れ防止や, ELP Lite[3] という身の回りの家電の電力消費量を可視化するハードウェアなどである. Nest Learning Thermostat[4] では, 部屋の温度を自動で調節し, 必要ないときは自動で判断し電気を節約する.

しかし, 個人での節電を支援する取り組みは, 会社や公共施設での取り組みと比べてその努力が目に見えにくい. ひとりでエレベータの利用を控えて階段を進んで使ったとしても, 取り組みが形に残らず, 節電に貢献している実感

が薄い. 家庭では, 節電の取り組みは最終的に電気代の請求書という形で本人に返ってきて, どれだけ節電しているか実感がわくものに対して, 階段の利用は証拠が残らない.

そこで, 本研究では公共の場における節電のみえる化を行い, 個人に対しどの程度の貢献ができたのかを可視化することでこの問題の解決を試みる. 日常の運動量から個人の健康状態を可視化, 向上することで, 家庭と同様に利益が発生する仕組みを考える. 個人が階段を使っているのか, エレベータを使っているのか, という状態を, Android 端末に掲載されるセンサを用いて, 利用者の状態を自動推定する Android アプリケーション NNCloud を開発する.

本研究の新規性は次のとおりである. (1) ユーザの四状態 (歩行, 階段, エレベータ, 停止) を自動判定するアルゴリズムを考案する, (2) ユーザが階段を使って節電した量を電力に換算して表示する, (3) 提案システムの実現可能性を示すため, 5日間の運用実験を行う. 本実験結果に基づき, 状態推定の精度と本システムで見積もられる電力量の信頼性を評価する.

2. 提案システム

2.1 NNCloud の概要

NNCloud は, ユーザがエレベータを使う代わりに階段

¹ 東海大学大学院
Tokai University Graduate School, Takanawa, Tokyo, Japan
² 明治大学
Meiji University, Nakano, Tokyo, Japan

を使った時間を測定し、節電の効果を確認可能にするシステムである。階段を歩行していることを Android に搭載されるセンサで自動推定することにより、利用者がエレベータを使う頻度を知覚させて、その利用を押さえ、節電を推奨する。逆にエレベータを使えば節電量が減っていくことで、エレベータの利用を控える。普段は記録にも残らないエレベータの利用を控えたことによる節電量を、明確な数値としてユーザに提示することで意欲の向上を図る。

2.2 階段歩行による節電量

階段歩行による節電量を次の様に見積もった。

まず、一般的な 15 人乗りエレベータ [5] の仕様より、消費電力を 12 kw とする。このエレベータを 1 階から 6 階まで昇降するのに必要な時間は、学内で計測したところ平均 20 秒であった。したがって、エレベータによる消費電力量 E は、

$$E = 12000 \frac{20}{3600} = 66.6[\text{W}]$$

である。これを、階段を使って 1 階から 6 階まで昇降したと仮定する。実測による経過時間は 90 秒であった。よって、以上の値から、階段を昇降するのに 1 秒当たり $E/90 = 0.74$ [W] 節電できる。逆にエレベータを利用する場合、1 秒当たり 3.33 [W] 消費する。

2.3 行動推定

行動推定は 5 秒間のサイクルで行われる。推定アルゴリズムは次のとおりである。

- (1) 歩数アルゴリズムを実行し、歩行の推定を行う。
- (2) 歩行していた場合は、次に階段の推定を行う。階段の条件を満たした時「階段」と出力、そうでない場合は「歩行」と出力する。
- (3) 歩行していない場合、エレベータの推定を行う。エレベータだった場合は「エレベータ」と出力、そうでない場合は「停止」と出力する。

2.3.1 歩数アルゴリズム

図 1 は、停止状態から 40 秒後、階段を昇り始め、その後 110 秒頃から歩行した時の加速度センサ値 dy と Roll[度] の変化を表している。加速度が 10 [m/s²] を中心に増減しているのがわかる。まず、加速度センサから得られる加速度 dy と、地磁気センサから得た値を使った端末の傾きを組み合わせ、地面と垂直な加速度を導き出す。その後、逐次入力された垂直加速度を最大値が更新されるたびに調べ、重力 $g = 9.8\text{m/s}^2$ を下回った際、今度は最小値を調べる。次に、重力 g を上回った際に最大値と最小値の差が閾値 0.98 を超えたか否かで「歩行」と判別する。

2.3.2 歩数の推定

歩数アルゴリズムで求めた歩数が閾値 θ を超えているかどうかを推定する。本方式では 5 秒間のセンサデータを推

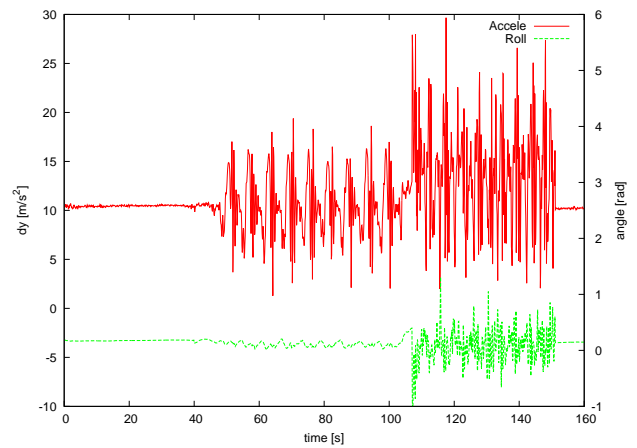


図 1 Z 軸加速度センサ dy と Roll の変化

定する。そのため閾値は一秒に一歩と考え、データの欠損を考慮し $\theta = 4$ 歩と設定した。

2.3.3 階段の推定

歩行と推定された場合、それが階段を歩いているのか、水平な道を歩いているのか切り分ける。本方式では、歩行と階段を切り分けるために Pitch と Roll の値に注目した。図 1 の加速度は歩行時も階段の昇降時も大きく変化している。しかし、Roll は階段を昇降している時に変動が小さかったのに対し歩行時は大きく変動しているのが確認できる。図 2 は歩行時と階段の昇降時の Pitch のヒストグラムである。図 2 では、階段の昇降が 100 度に偏っているのに対し、歩行はばらつきが大きい。これらのことから、Pitch と Roll の分散の合計 σ^2 が閾値 $\theta_\sigma = 800$ を下回った際に「階段」と推定する。

2.3.4 エレベータの推定

図 3 にエレベータに乗車時の Z 軸方向の加速度の変化を示す。発進時と停止時に大きな加速度の変化が確認できる。この特徴を微分し、それらの最大値と最小値から閾値 80% 以内の値を探す。見つけた値と対応する逆の特徴、正の値を見つけた場合は負の値を、負の値を見つけたときは正の値を探し、それぞれ特徴が閾値 1.9s 以内だった場合エレベータと推定する。また、エレベータは発進時と停止時以外の時は特徴が現れないため、エレベータが発進した特徴を推定したのち停止状態だった場合もエレベータと推定する。

2.4 実装

提案システムのクライアントを Java 言語を用いて、Android 上に実装した。バージョン 2.2 から 4.2 までに対応している。

端末はどのような向きで携帯されるか分からない為、常に地面から垂直方向の加速度の変化を読み取るように X, Y, Z 方向のセンサの値を正規化する必要がある。これには、加速度センサと地場センサを用いた *1。

*1 従来広く用いられていた ORIENTATIN センサは、現在では失

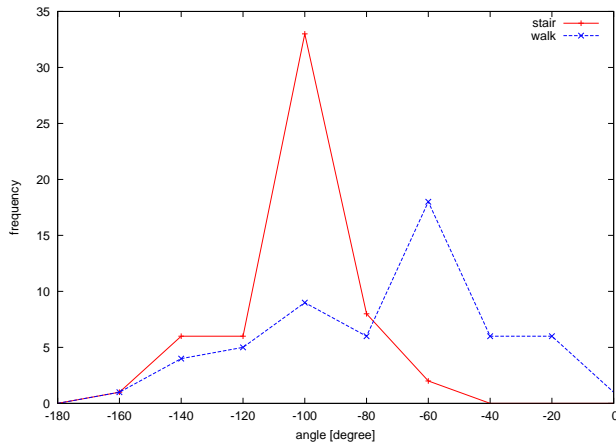


図 2 階段と歩行時の Pitch ヒストグラム

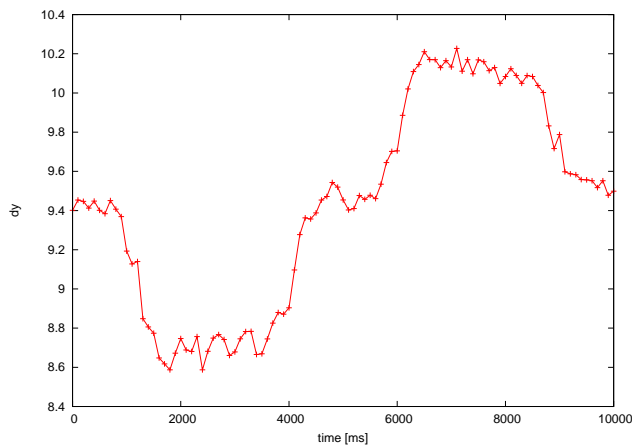


図 3 エレベータ内の Y 軸加速度の変化

状態を記録する為に、大量のセンサデータを逐次格納する必要がある。この目的の為に、データベースに格納し、必要に応じて取り出すことを考えた。しかし、データベースはセンサデータを書き込み続ける中取り出さなくてはならず、必然的に I/O が足りなく処理が遅いことが分かった。そのため ArrayList を利用して格納し、終わったらオブジェクトを削除する方式にした。

図 4 は作成した NNCloud の実行例である。現在の状態を表すフィールド（上）とその一日の行動を振り返ることを可能にした履歴のフィールド（下）の二つから構成されている。図の例では、現在状態は「停止」で、その日の累積歩行数 544 歩であることを示している。履歴フィールドの操作は、上下フリックでスクロールするとその一日をさかのぼり、左右フリックで過去の日付への移動ができる。図の例では、5 月 14 日に 2,555 歩の歩行したことを表している。

サーバとの通信は POST/HTTP を用いる。サーバは HTTP サーバで php で作成した API により、ユーザの登

録された deprecated method に分類されている。ORIENTATION センサは実際には加速度センサと地場センサを複合して実装されているので、今後のメンテナンスを考え加速度センサと地場センサの値から直接端末の傾きを計算している



図 4 NNCloud の実行例

録、ログイン、マイルの送信、ログの送信を提供する。ユーザ登録はランダムで一意的な疑似 ID とパスワードを発行することによって行われる。ユーザはサーバとの通信を意識することなく利用することができる。

サーバとの通信は 1 時間に 1 度行う。もし接続に失敗した場合は次のタイミングで接続時に再送信する。

3. 評価

3.1 実験 1: 状態推定精度

提案方式の精度を次の方法で評価した。2013 年 3 月に、5 名の被験者に、端末を持った状態で大学構内で、歩行、階段の昇降、エレベータの乗車を順に行ってもらい、センサの値を測定した。エレベータの一往復の時間が 40 秒であったため、被験者は各状態を 40 秒測定することとした。

判定結果を表 1 に示す、入力データの単位は秒である。平均適合率は 0.58 である。

表 1 提案方式の精度 [s]

真 \ 判定	walk	stair	elevator	stop	sum
walk	165	35	0	0	200
stair	115	80	0	5	200
elevator	15	25	110	50	200
stop	0	15	80	105	200
適合率 p	0.83	0.4	0.55	0.53	
平均 \bar{p}					0.58

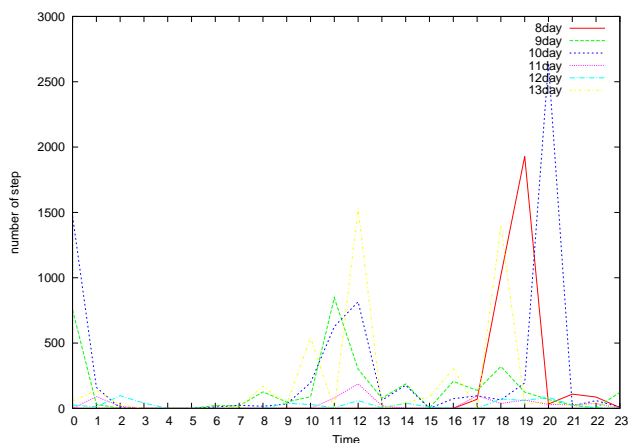


図 5 歩数の時間分布 (u_7)

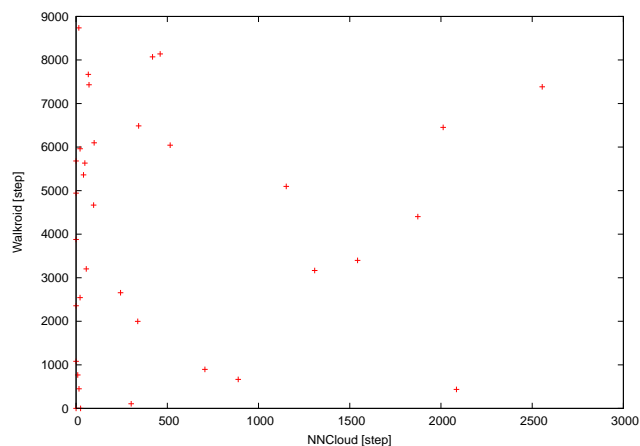


図 6 NNCloud と万歩計アプリ Walkroid の歩数散布図

3.2 実験 2: 運用実験

長期間での使用に問題なくユーザの動作が測定できるか、どこまで節電できるかを調べるため以下の実験を行った。7人の被験者を募り2013年5月8日から14日まで運用実験を行った。被験者は大学生と教員である。

実験1によりエレベータが停止時に誤推定される割合が高いことが分かったため、エレベータを推定せずに実験を行うこととした。歩数のアルゴリズムの誤差を補うために、「万歩計」AndroidアプリのWalkroid[7]を利用した。被験者には終了時Walkroidの歩数とNNCloudの歩数それぞれを申告してもらった。

図5に、ユーザ u_7 の歩行量の時間分布を表す。ウィークデイである8,9,10日に関しては、通学時の11時頃と帰宅時の19時前後に大きなピークが見られる。

図6は、全被験者の二つのアプリNNCloudとWalkroidの測定値の散布図である。Walkroidが多くの場合NNCloudより高い歩数を測定しているが、逆のケースもいくつか見られる。アルゴリズムの精度には個人差があることが示されている。

表2に、NNCloudで推定した状態の統計量を示す。一日のうち、9割以上が「停止」であり、活動している期間は短い。

	平均	標準偏差	割合
stop	29398	20451	0.9814
walk	262.04	470.66	0.0087
stair	295.10	561.42	0.0099

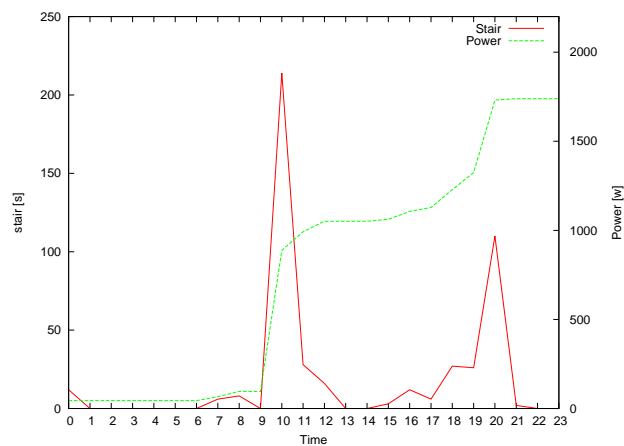


図 7 階段歩行時間と累積節電力量

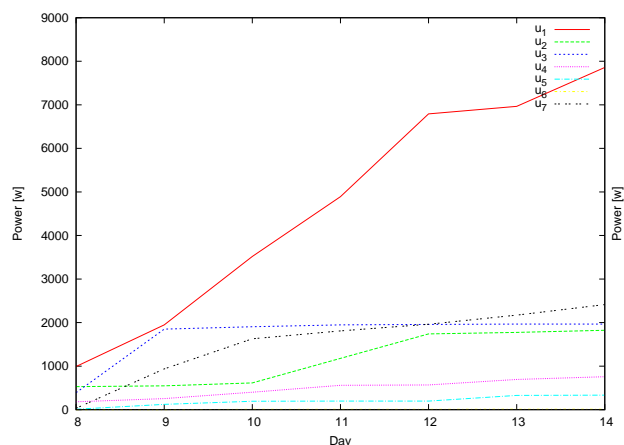


図 8 ユーザ毎の累積節電力量

3.3 節電効果

階段の利用量を電力に換算して全ユーザの総節電量を調べる、図7では、利用者が10時ごろ出かけたのち、20時ごろに帰宅した際に利用していた階段の昇降時間とそれによる節電力量の関係を表している。ユーザが階段を使うことで節電量が増加していることが確認できる。

図8は、本実験で全ユーザの節電の節電力量である。ユーザ u_1 が最も多く節電量を記録している。ユーザ間でこのような節電量を比較することで、節電の意欲が向上することを期待している。

以上の結果より、本実験では、7名1週間で15,160 [W]節電できたことが明らかになった。

3.4 考察

実験1の表1より、停止時または階段昇降時にエレベータと誤推定してしまうことが多いことが分かった。階段昇降時に誤推定されるのは、垂直加速度の計算に誤りがあり、端末の向きによっては重力加速度がマイナスになってしまうためである。停止時の誤推定は、端末の微弱な振動を特徴量としてとらえてしまい、エレベータと誤推定してしまっていたことが原因である。

実験2の図8では被験者 u_1 が大きく節電量を記録している。これは u_1 が意識して特に歩き回っていたことと、端末の持ち方が一定で歩行アルゴリズムが正しく動作できていたためである。Android 端末の持ちが悪かったり、鞆に入れて持ち歩いたり、個人差が大きいことが分かった。それぞれで歩行の判定しきい値を学習するような工夫が今後必要である。

4. おわりに

利用者の状態を自動推定し、階段を歩いて節電した量を自動計測するシステムを開発し、その基本機能の確認と運用実験の結果を報告した。

NNCloud のアルゴリズム改善を行い、ユーザ数を増やした長期的な運用実験を今後の課題とする。歩数アルゴリズムは垂直加速度の計算式の修正に加えて、微弱な加速度のスキップを行うエレベータ判定方式の改良が必要である。更に、推定状態を増やし、エスカレータ、自転車、自動車などをも節電の対象として広げていきたい。ポータルサイトを作成しユーザの行動や、ほかのユーザの節電力量などを確認することも計画している。

参考文献

- [1] パナソニックの節電取り組み入手先 (<http://panasonic.co.jp/eco/setsuden/>) (2013 03 28 参照).
- [2] 政府の節電ポータルサイト入手先 (<http://setsuden.go.jp/>) (2013 03 28 参照)
- [3] ELP Lite available from (<http://www.sassor.com/>) (2013 03 28 参照)
- [4] Nest Learning Thermostat 入手先 (<http://greenz.jp/2011/11/24/nest-learning-thermostat/>) (2013 03 28 参照)
- [5] 三菱電機 エレベーター・エスカレーター AX-IEZ: 乗用 [P] 機種一覧・据付図(寸法) 入手先 (<http://www.mitsubishielectric.co.jp/elevator/product/axiez/axiez-p/model.html>) (2013 03 28 参照)
- [6] 楊 天輝, 梶 克彦, 河口 信夫, “加速度センサを用いたエレベータ区間検出と移動距離推定”, 情報処理学会第74回全国大会, pp. 441-442, 2012.
- [7] Walkroid - simple pedometer 入手先 (<http://walkroid.kino2718.net/>) (2013 03 28 参照)
- [8] Gatanne Hach, , Edward D. Lemaire, Natalie Baddour, “Wearable Mobility Monitoring Using a Multimedia Smartphone Platform”, IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL. 60, NO. 9, pp.3153-3164, 2011.
- [9] 大久保成晃, 菊池浩明, 各種センサを統合した行動状態推定機能を持つライフログシステムの Android への実装と評価, 情報処理学会第74回全国大会, 6X-2, Vol.3, pp. 35-36, 2012.