

SUWANT!:

電車で効率よく座るための支援アプリケーション

笹川 真奈^{1,a)} 椎尾 一郎^{1,b)}

概要: 日本の都市部では、多くの人々が長時間の電車通勤・通学を強いられている。その通勤時間に、電車の中で座って休めるか否かは重要問題である。本研究では、電車内で乗り合わせた乗客が降りる駅を、過去の事例から推定・提示することで、席に座れるチャンスを利用者に提供するスマートフォンアプリケーション SUWANT!^{a)}を提案・実装した。本アプリケーションは、スマートフォンや各種デバイスが発信する Bluetooth 信号を利用して、電車内の乗客を特定する。そして Bluetooth 機器所有者が過去に乗降した駅のデータから、特定した乗客の降車駅を推定する。もしユーザの周辺にすぐに降りそうな乗客がいれば、本アプリケーションはその乗客の存在と推定距離と推定降車駅を表示する。ユーザは座っている乗客の周辺を歩き回り、変化する推定距離を見ることによって、もうすぐ降りそうな乗客を探し当てることができる。また、本アプリケーション使用中にも Bluetooth 信号を記録する。そして周辺の Bluetooth 機器所有者の乗降駅をデータベースに登録するためのデータを、サーバに送信する。本アプリケーションのユーザは、アプリケーションを使えば電車で座れる可能性が上がるという動機で、Bluetooth 信号の情報を収集するクラウドソーシングに参加することになる。評価実験では、特定の電車区画において、10 乗車中 2 回本システムを使って座ることができた。

^{a)} SUWANT! は “Suwaru” (座る) と “want” (したい) を組み合わせた造語である。

キーワード: スマートフォンアプリケーション; Bluetooth; クラウドソーシング; 電車;

SUWANT!:

Support Application for Seat Availability Determination on Trains

SASAGAWA MANA^{1,a)} SHIO ITIRO^{1,b)}

Abstract: In the modern Japanese commuting scene, most Japanese people desperately want to find seats on the train to get some rest on their long commute. SUWANT!^{a)} is a smartphone application using Bluetooth that helps users on a train to find passengers sitting nearby who will disembark at the upcoming stations; this applications is powered by crowdsourcing. The users of this application are motivated to contribute to crowdsourcing by being rewarded with an available seat. Upon startup, SUWANT! displays on the phone screen whether there are passengers around the user who are likely to disembark soon, and the distance between these passengers and the user. The user searches for such passengers by hovering around sitting passengers and seeing the changes on the screen. On an experimental test, SUWANT! has succeeded in showing potential vacant seats once in every 5 rides on a specific train.

^{a)} SUWANT is a combination of “Suwaru” (take a seat, in Japanese) and “want.”

Keywords: Smartphone application; Bluetooth; Crowdsourcing; Train;

1. はじめに

電車は主に都市部に住む日本人にとって重要な交通手段である。なかでも東京都では、大多数の人々が電車を通勤手段としている^{*1}。また、日本人の平均通勤時間は1時間である^{*2}。このように、電車への依存と長い通勤時間を考えると、可能ならば電車で座席に座りたいと希望するのは、筆者らだけではないと考えている。しかし、電車の座席に座れるかどうかは完全な運である。例えば、すぐに降りる乗客の近くに立っていれば早く座ることができる一方で、自分よりも後に乗車してきた乗客がたまたますぐ降りる乗客の前に立ってしまい、自分よりも先に座ってしまうこともある。それゆえ、電車で座って休みたいと考えている多くの人々のために、電車ですぐに降りるであろう乗客を探すスマートフォンアプリケーションを提案・実装した。本アプリケーションは、図1のように電車内にてBluetooth信号を用いて乗客の情報を集め、集めた情報を元にクラウドソーシングにて周辺乗客の降車駅を推定する。

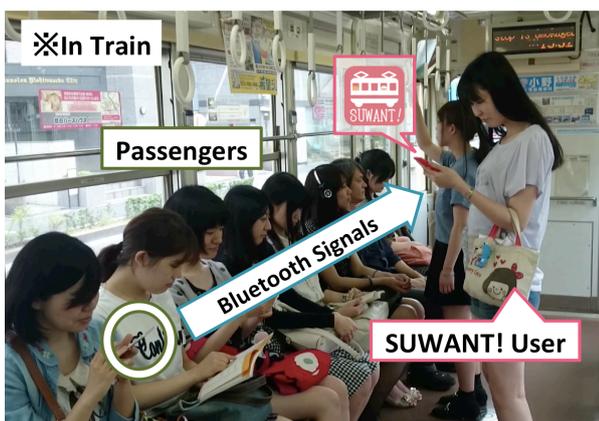


図1 SUWANT!使用場面のイメージ図

近年では、半数以上の日本人がスマートフォンを所持している。ほぼ全てのスマートフォンにはBluetooth通信機能が内蔵されており、様々な周辺機器との接続にBluetoothが用いられている。さらに、ヘッドフォンやスマートリストバンド、スマートウォッチ等の、Bluetooth接続ウェアラブル機器が普及してきたことにより、BluetoothをONにした状態のスマートフォンを所持する乗客の数は今後さらに増加するであろう。また、対象端末が半径10m程度以内であって、対象端末のBluetoothが検知可能モードになっていれば、その端末の独自のBluetooth IDと電波強度(RSSI)を取得することができる。そこで、Bluetooth信号を用いて乗客の行動を推定できると考え、本システムを設

¹ お茶の水女子大学
Ochanomizu University, Bunkyo, Tokyo 112-8610, Japan
a) g1120520@is.ocha.ac.jp
b) siiio@acm.org
^{*1} <http://www.stat.go.jp/data/kokusei/2000/jutsu1/00/04.htm>
^{*2} <http://www.mlit.go.jp/common/001001523.pdf>

計した。すなわち、乗客のBluetooth IDが停車駅で検知、消失することを記録すれば、様々なBluetooth機器所有者の乗降駅を特定することが可能である。さらに、このデータを大量に用意し、車両内のBluetooth IDと各々のRSSI値を追跡すれば、周辺のBluetooth端末所持者の車内位置と乗降駅を特定することができるであろう。

本方式では、より多くの乗降駅情報を集めるほど、ユーザの周辺乗客の乗降駅を正確に提示できる確率が増えてゆく。しかし、膨大な量の様々なデータを一人で集めるためには多大な時間がかかる。そのため、本アプリケーションを利用する多数のユーザでデータを集め、これを全ユーザで共有すれば、より早く座れる確率を高めることができる。そこで、図2に示すような、各ユーザが集めた様々な電車区間・時間の乗客情報をサーバに集約するクラウドソーシングの手法により、乗客の乗降駅データベースを構築する。これにより、ユーザが増えれば増えるほどもうすぐ降りる乗客を探し当てられる確率が増えていくであろう。情報収集作業には、本アプリケーションをONにするだけで、手間をかける必要なく参加できる。また、本アプリケーションを使えば電車に座る可能性が上がるという動機付けにより、ユーザは積極的に情報収集に参加することになるであろう。

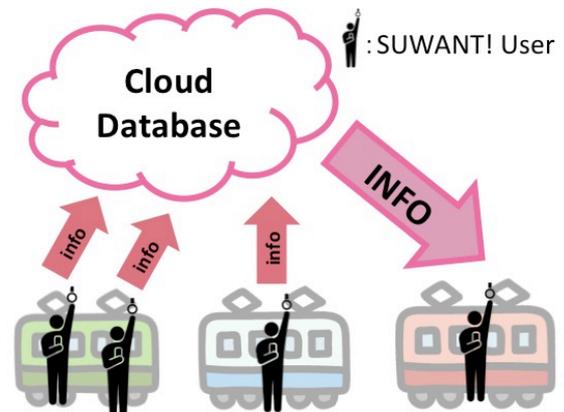


図2 SUWANT!クラウドソーシングのイメージ図

2. 実地調査

2.1 電車内 Bluetooth 取得数の現状

SUWANT!の仕組みは、Bluetoothが検知できる状態にある端末を所持した乗客が、同じ車両に多数乗り合わせている前提で機能する。そこで、実際に運行されている電車内にBluetoothが検知できる端末がどの程度存在しているのかを調査した^{*3}。調査のために、12秒毎にBluetoothIDを取得し、取得時間と共に随時サーバにアップロードするアプリケーションを実装した。調査は、時間帯を5つ(通勤/午前/午後/夕方/帰宅)に分類し、各々の時間帯ごとに

^{*3} 測定にはXperia Z1を用いた。

前方・中方車両にて2周ずつ、計20周、JR東日本の山手線に乗り、Bluetooth IDを取得した。一駅間の平均取得Bluetooth ID数を図3に示す。

この結果、1車両に平均して1-3程度のBluetoothIDを常に検知することができた。乗車位置周辺で最低1人でも乗降駅が分かれば降車情報を表示できるので、現時点でもSUWANT!は実際に運行されている電車内にて機能すると考える。

時間帯・車両別 1駅間 平均取得BluetoothID数											
	通勤 (7:32~)		午前 (10:16~)		午後 (13:20~)		夕方 (15:45~)		帰宅 (18:39~)		車両別 平均
前方車両 (2両目)	1.0	0.7	2.6	1.3	1.2	3.1	3.2	0.7	0.9	1.6	1.6
中方車両 (5両目)	1.1	1.6	1.2	2.3	2.0	1.2	1.6	1.9	1.4	2.2	1.7
時間帯別 平均	1.1		1.9		1.9		1.8		1.5		1.6

図3 山手線20周実験結果(1駅間の平均取得Bluetooth ID数)

2.2 RSSI値と距離の関係

SUWANT!は、まもなく降車すると推測された乗客までの距離を、対象乗客のBluetooth信号のRSSI値から推定して表示する。RSSI値は周囲の環境の影響を大きく受けると予想されるので、実際の電車車両内でRSSI値と距離の関係を調査した*4。

2.2.1 混雑度別：RSSI値と距離の関係

BluetoothのRSSI値は、車内の混雑度によっても変化すると予測される。そこで、表1に示すように混雑度を3段階に分類し、各々の混雑状況において次の測定を行った。すなわち、検知可能Bluetooth端末を所持している被測定者が、図4に示すように、1.(目の前)、2.(ブロック内)、3.(ブロック外)、4.(別車両)に存在するとき、そのRSSI値を各々10回ずつ測定し、RSSI値と距離の関係を調査した。この結果を表2に示す。これから、混雑度によって同じ距離でも受信するRSSI値が異なることがわかった。

表1 混雑度分類

混雑度	乗車率	様子
低混雑度	0%から100%	空いている席がある
中混雑度	100%から200%	席は埋まり立っている人が数人
高混雑度	200%以上	歩き回るのは厳しい

2.2.2 中混雑度：RSSI値と距離の関係

SUWANT!のユーザは、これから降車しそうな乗客の前に移動して席を確保する。極度に混雑した車両では、移動が困難なため本システムを使用しても席の確保は難しくなる。一方で空席のある空いた車両で本システムが使用され

*4 調査では送受信ともXperia Zを用いた。

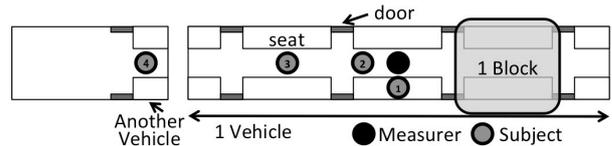


図4 混雑度別実験での測定者・被測定者の配置図

表2 混雑度別：10回測定平均RSSI値(dBm)と距離の関係

被測定者位置	低混雑度	中混雑度	高混雑度
1. 目の前	-60	-63	-65
2. ブロック内	-68	-80	-91
3. ブロック外	-86	-90	0
4. 別車両	0	0	0

ることは無い。そのため、本システムが使用される車両の混雑度は、中混雑度が主であると考えられる。そこで、中混雑度での距離とRSSI値の関係を、距離間隔を細かくして再調査した。図5で示すように、検知可能なBluetooth端末を所持している被測定者を実際の電車の座席に座らせ、被測定者の座席の位置から1席ずつ離れて立ちながら、各座席の前で計10回ずつ被測定者のBluetoothのRSSI値を測定した。

図6にこの測定結果の平均値を示す。実際の測定値は、この平均値の上下に分散している。そこで、表3に示すように閾値を設定して、対象乗客への距離を提示することにした。実際には、図7などに示す画面例のように、降車すると推定される乗客への距離を、座席のアイコンで表示する。

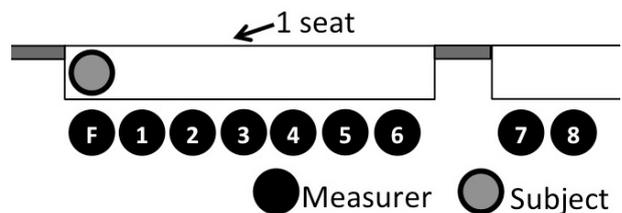


図5 中混雑度実験での測定者・被測定者の配置図

Average of ten measurements of Bluetooth RSSI (dBm)									
Distance from the user	Directly in front	1 seat	2 seats	3 seats	4 seats	5 seats	6 seats	7 seats	8 seats
Average RSSI (dBm)	-62.8	-73	-72.68	-73.5	-75.6	-79	-75.8	-87	-85.1

図6 中混雑度：10回測定平均RSSI値(dBm)と距離の関係

3. SUWANT!

3.1 画面説明

SUWANT!は、もうすぐ降りるであろう周辺乗客を探し当てることにより、より早く席に座れる確率を高めるためのスマートフォンアプリケーションである。本アプリケー

表 3 SUWANT!における RSSI 値と距離の関係

取得 RSSI 値	推定されるユーザと対象乗客との距離
-64dbm 以上	目の前にいる
-65dbm から -74dbm	1 から 3 席分離れている
-75dbm から -84dbm	4 から 6 席分離れている
-85dbm 以下	7 席分以上分離れている

ションは、周辺の乗客が所持する機器の Bluetooth 信号を検出し、クラウドソーシング手法を用いて収集した降車駅データベースにより、電車の乗客の乗降駅を推定・提示する。

本アプリケーションの初期画面を図 7 に示す。図 7 は、本アプリケーションが具体的な値を取得し表示する前の、指標項目のみを表示している。本アプリケーションの画面は、画面左下を原点とした 2 次元グラフのように見る。すなわち、原点から右方向の横軸、x 軸が対象乗客とユーザ間の距離、縦軸の y 軸が対象乗客の推定乗降駅であり、この 2 次元座標空間の点が、対象乗客の端末機器の種類である。

左下の四角に囲まれた部分に現在駅(原点位置)が、左上の横長のボタン上に現在路線が表示される。画面の左側(y 軸)には、次に来る駅が、現在駅(原点位置)から順番に並べて表示される。つまり、図 7 で左側のリストの一番上に表示されている駅が、現在駅から 3 駅目の駅である。これらの駅と現在路線は、ユーザの現在位置情報から推定されており、電車が動きユーザの現在位置が変化すると、連動して更新される。並行する別路線がある場合に、推定結果の現在路線が誤って表示されることがある。その場合、現在路線が表示されているボタンをタップすることにより、ポップアップメニューから正しい現在路線をユーザが選択することができる。また、電車の進行方向が誤って検出されている場合は、左上の“↑↓”のボタンを押すことにより正しく変更することができる。これらにより、次に来る駅が正確に表示されるようになる。画面下(x 軸)では、ユーザからの距離の指標をピクトグラムで示している。一番左端に赤色で表示された立っている人のアイコンは、本アプリケーションのユーザを示している。

ユーザが電車に乗り SUWANT! を ON にすると、アプリケーション上の画面が、ユーザとデータベースの状況によって変化し続ける。もしユーザの周辺(座席 8 席分以内)に、来たる駅でもうすぐ降りる乗客がいると本アプリケーションが推定すれば、図 7 D) のようなアイコンが出現する。アイコンの形状は、対象乗客が所持している Bluetooth 端末機器の種類、例えば、iPhone、Android、ヘッドフォン、携帯電話等を示している。図 7 では Android Robot アイコンが表示されていて、ユーザの周辺に Android 端末を所持している乗客がいることを示している。対象乗客が画面上に示されている種類の端末を使用していることが明確に確認できれば、このアイコンにより対象乗客を特定する

ことができる。アイコンの横方向位置は、ユーザと対象乗客の距離に応じて変化する。例えば、もしユーザがアイコンで示される対象乗客に近づけば、アイコンは左に動き、遠のけば右に動く。また、アイコンの縦方向位置は電車の進行により変化し、駅名と連動して下に移動する。

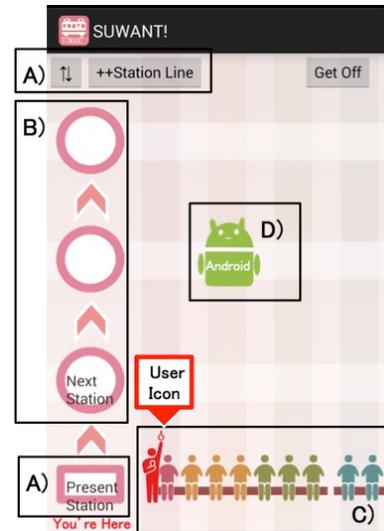


図 7 SUWANT!初期画面：A) 現在路線、駅名 B) 来る駅指標 (y 軸) C) ユーザからの距離指標 (x 軸) D) 端末機器の種類

3.2 使い方

SUWANT!の使い方を、実際のアプリケーション画面のスクリーンショットである図 8 を用いて、以下で説明する。

- (1) 電車に乗車した際、SUWANT!を起動する。
- (2) SUWANT!は、図 8 に示すように、周辺の Bluetooth 端末機器(この例では 4 台)を検知し、情報を画面上に提示する。例えば、上の灰色のリングマークのアイコンは、以下の情報を示している。
 - A). iPhone を使っている乗客が周辺にいる。
 - B). 対象乗客は、三駅目の荒川車庫前の駅で降りると推定される。
 - C). ユーザは、対象乗客から 4-6 席分程離れている。
- (3) ユーザは、アイコンが示す対象乗客が周辺乗客のうち誰なのか、を特定する。一つの手段としては、変化するアプリケーション画面を見ながら、電車内を歩き回ることである。電車内を歩き回れば、ユーザと対象乗客との距離が変わり、アイコンが左右に移動する。もし、ユーザが対象乗客の目の前に移動すれば、対象乗客のアイコンが一番左端、つまり赤のユーザアイコンの真上に移動するはずである。
- (4) 対象乗客のアイコンが左端にきた時、目の前にいた乗客が探し求めていた対象乗客となる。
- (5) ユーザは対象乗客の前に立ち、対象乗客の推定降車駅まで待っていれば、対象乗客が降りた後に席に座るこ

とができる。

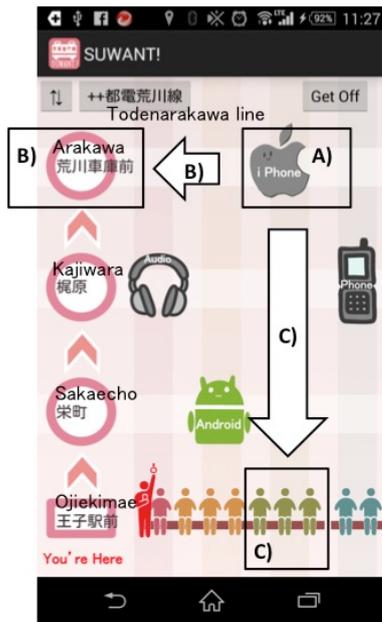


図 8 SUWANT!画面例：A).iPhone 所持 B). 荒川車庫前で降りる C). ユーザから 4-6 席分離れている

4. 実装

本アプリケーションは、Android OS 搭載スマートフォンアプリケーションとして、Java 言語で実装した。また、クラウドサービス Azure 上に仮想サーバを立て、サーバサイドのシステムを php 言語と MySQL で実装した。本アプリケーションは、起動すると 12 秒毎に周囲の Bluetooth 情報 (ID と RSSI) を取得し続ける。Bluetooth 情報は Android の標準 API を用いて取得する。この API を用いることにより、周辺乗客が持っている Bluetooth 機器、例えばスマートフォンやヘッドフォン、ノート PC、携帯電話等の Bluetooth 情報を取得することができる。Bluetooth 情報を取得する一方で、取得した情報を用いた以下の二つのタスクを実施する。

4.1 情報収集

本システムは、クライアント (スマートフォン) でユーザの周囲にある Bluetooth 機器の情報を収集し、サーバで乗客の乗降駅データベースを構築する。図 9 に、これを実現するシステム構成図を示す。

4.1.1 クライアントサイド

クライアントサイドであるスマートフォン上では、検知した Bluetooth 情報をローカルのテキストファイルに書き込み、ユーザが電車を降りる際に、このファイルをサーバにアップロードする。アップロードする情報とは、検知した Bluetooth 情報 (ID と RSSI)、現在位置、タイムスタンプ

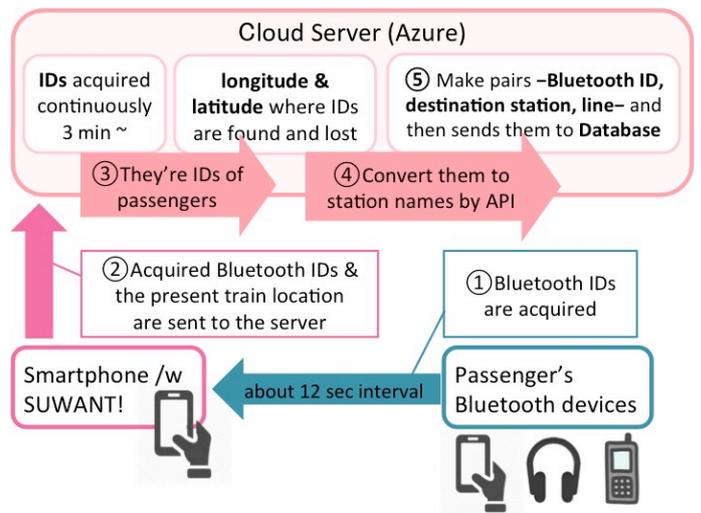


図 9 システム構成図

の 3 種類である。現在位置は、Android の標準 API によって取得された緯度経度である。

4.1.2 サーバサイド

一方、このファイルを受け取ったサーバでは、取得した情報から乗客の乗降駅を推定するためのバッチ処理を 1 日 1 回行う。この処理を以下で説明する。

(1) 前処理

まずは、生データからノイズを取り除くため、2 段階の前処理を行う。最初の処理は、Bluetooth 信号を連続して取得できなかった場合の処理である。Bluetooth 信号の受信はしばしば不安定であり、同じ車両に乗っている乗客であっても、その所有する機器の Bluetooth 信号が連続して取得できないことがある。この場合、もしその乗客が電車に乗り続けていれば、再び検知できる可能性が高い。そこで、1 時間以内に再検知できた Bluetooth ID は、同じ車両に乗り続けていた乗客の機器の ID として処理をする。閾値を 1 時間に設定した理由としては、東京都の平均通勤時間が 1 時間であり、同じ車両に 1 時間以上乗る乗客は少ないと判断したためである。通勤形態の異なる他都市でのデータを処理する場合には、閾値を調整する。次の処理は、Bluetooth 信号が短い間しか取得できなかった場合の処理である。車内のスマートフォンは、線路沿いの民家から発せられている Bluetooth 信号、すれ違った電車やホームにいる乗客の Bluetooth 信号を受信することがある。そこで取得継続時間が短い信号は、情報取得者と同じ電車の車両に乗っている乗客ではないと認識し除外する。実際には、3 分以上連続して取得できていない Bluetooth ID を分析対象から除外している。東京都での駅間が 3 分以上であるため、閾値を 3 分に設定したが、他都市でのデータを処理する場合には、最初の処理と同様、閾値を調整する。

(2) 本処理

ノイズを取り除く前処理を終えたら、次に各 Bluetooth ID の端末所持者の乗降駅を推定する。ここでは、Bluetooth

ID が検知され始めた地点を、その Bluetooth ID の端末所持者が電車に乗ってきた駅と判断する。また、連続で取得できていた Bluetooth ID が検知できなくなった地点を、その所持者が電車から降りた駅だと判断する。各 Bluetooth ID が出現・消失した地点は、webAPI*5 を用いて、緯度経度から路線名、駅名に変換する。また、乗客が往路に乗車した駅は、その乗客が復路に降車する駅と考えられるため、乗車駅、降車駅はいずれも同じ情報として扱う。この情報から、- Bluetooth ID, 降車駅名, 路線名 - の組を作り、データベースに登録する。本システムのユーザが増えれば増えるほど、データベース上のレコードが増えていく。これによりデータベースがより正確になり、もうすぐ降りる周辺乗客を推定しやすくなるであろう。

4.2 情報提示

クライアントアプリケーションは、情報収集にて集められたデータを用いて、ユーザの周辺乗客の中で来る駅でもうすぐ降りそうな乗客を推定し提示する。周辺乗客の降車駅は Bluetooth ID で推定し、ユーザとの距離は RSSI の強さを用いて推定する。

クライアントアプリケーションが Bluetooth 信号を検知すると、サーバに対して、取得した Bluetooth ID と現路線名に対応した駅名が登録されているかどうかの問い合わせを行う。もしデータベースにその Bluetooth ID と路線名が含まれた組が登録してあれば、サーバはその降車駅名を返す。その際、複数の降車駅が登録されている可能性があるが、クライアントアプリケーションは、ユーザの画面上に表示されている駅名(図 7 B)) に該当する情報のみを表示する。複数の降車駅名がデータベースに登録されていた場合、現在の実装では、クライアントユーザが乗車している路線にある駅名に該当するすべての駅名を返す。将来的には、より有用な降車駅のみを返すようにしたいと考えている。例えば、対象乗客の駅ごとの降車頻度や、時間帯、曜日等によってもフィルタリングをしたいと考えている。

クライアントが降車駅を受け取ると、その Bluetooth 端末所持者までの距離を RSSI 値を用いて推定する。そして、第 2 節で示したように RSSI 値を元にした推定距離を表 3 に基づいて表示する。しかしながら、関連研究 [1] にも示されている通り、RSSI 値だけで正確に推定するのは困難である。そのため、ここで提示する情報はあくまで目安であり、ユーザはその目安を元に、対象乗客が目の前にいると RSSI 値で判定できるまで、対象乗客を歩いて探し回る必要がある。

5. 評価実験

本アプリケーションを、- Bluetooth ID, 降車駅名, 路

線名 - の組の取得効率と、席に座れた成功率の二つの面から評価する。

5.1 ID-降車駅情報の取得効率

- Bluetooth ID, 降車駅名, 路線名 - の組の情報を取得する効率を測るため、20 人のユーザに日常の電車使用時に本アプリケーションを使用してもらった。結果、表 4 のような結果を得た。

各ユーザごとにデータを収集し始めた時期が異なるため、使用日数期間は 2 日間から 67 日間とばらばらであった。全ユーザの使用日数を合わせると述べ 392 日となり、全ユーザが集めた ID-降車駅名-路線名の組数を合計すると 1675 組であった。各ユーザは 1 日往復で 2 回電車に乗ることを考えると、平均して、1 回乗車するごとに約 2 組のデータが得られることがわかる。

表 4 20 人のユーザが集めた組数

ユーザ番号	使用期間(日)	取得組数	1日の平均取得組数
1	25	77	3
2	9	9	1
3	13	129	10
4	7	3	0
5	15	6	0
6	47	473	10
7	4	12	3
8	16	44	3
9	67	382	6
10	13	150	12
11	21	64	3
12	48	289	6
13	6	11	2
14	2	13	7
15	40	187	5
16	10	44	4
17	6	32	5
18	25	39	2
19	13	53	4
20	13	150	12

5.2 成功率

本アプリケーションを用いて、実際どのくらい座れたかを検証した。筆者のうち一人が、表 5 で示す条件下において、本アプリケーションを用いて座れるかどうかを実際に試してみた。これらの電車は、席は埋まっているが適度に車内を歩き回れる混雑度であった。しかし、都心部へ向かうため、電車が進むにつれ混雑度が上がり空席を得ることが困難になる。また駅間が他の路線よりもやや長い間、座れるチャンスがくる頻度が少ない状況下であった。そのため、本アプリケーションの利用で恩恵を受ける状況として適していたと考える。何故なら、このような状況下にお

*5 <http://express.heartrails.com/api.html>

いては、座りたいと感じる人や、座れたら嬉しいと強く思う人がより多くなると考えているからである。また、本検証を行う前に、検証を行う区画内で降車すると推定されている乗客のデータ、つまり - Bluetooth ID, 降車駅名, 路線名 - のレコードを、100 組以上データベースに収集しておいた。結果として、本アプリケーションを使用することで、10 回乗車して 2 回、座ることができた。そのため、使用区間のデータが 100 組以上あれば、20% の確率で座れるチャンスが来ることになる。第 5.1 節で述べたように、1 回乗車すると平均して 2 組の新しいデータを得ることができる。このため、100 組以上のデータを集めるためには、使用したい区画において 50 人以上のユーザがいれば、1 日で集めることができる。

表 5 検証時の乗車電車条件

路線名	出発時刻	区画
京王線	2015/2/16-20 8:52	仙川から新宿
中央線	2015/2/16-20 10:16	立川から新宿

6. 関連研究

電車で快適に過ごすためのスマートフォンアプリケーションをいくつか紹介する。“トレインネット”^{*6} は、東京の山手線において、車両の重量から推定される車両ごとの混雑度を表示するアプリケーションである。このアプリケーションを活用することによって、これから乗る電車のうち、一番空いている車両に乗ることができる。また、“こみれぼ”^{*7} は、様々な路線の車両の混雑度を、口コミによって共有するサービスである。このアプリケーションを使用することによって、より空いている電車に乗るために、乗ろうと予定していた電車の前後の電車に乗り換えたり、他の路線を選択したりすることができる。この 2 つのアプリケーションは、より空いている車両に移動することができるので、座れる確率を高めることができる。しかし、もうすぐ降りる乗客を示すわけではないため、本アプリケーションの方が座れる確率をより高めることができる。

また、Bluetooth 信号のみを用いて、人の行動や位置を推定している研究は、[2] で紹介されているように多数ある。牛越ら [3] は、外部環境中にある固定局からの Bluetooth の検知パターンによってユーザのライフログを推定した。田岡ら [4] は、固定局を用いて Bluetooth 信号による室内の位置推定の手法を提案した。本研究では固定局を用いず、ユーザや他乗客の所持するスマートフォンが発する Bluetooth 信号のみを使用している。また前川ら [1] は、電車での乗客のスマートフォンのみを用いて、車両内の混雑度と、乗客が車両内のどの位置にいるのかの推定を行った。本研究では、スマートフォンの Bluetooth 信号のみを用い

た位置推定が不安定であることを、実際のユーザが移動して確認する行動によって補完しようとした。

7. まとめと今後の課題

本論文では、Bluetooth とクラウドソーシングを用いて、電車の乗客の行動を推定し、もうすぐ降りる乗客を見つけ出し、より早く席に座れる可能性を高めるスマートフォンアプリケーションを開発した。開発したアプリケーションは東京都の特定の電車区間で使用し、10 回アプリケーションを使用したうち 2 回座ることができた。今後、本アプリケーションを用いた検証をさらに進めていきたい。また、積極的にユーザを増やし、データベースのデータの量も種類の幅も増やしていく予定である。さらに、生データから必要なデータへと分析する際に、より有効なアルゴリズムを検討したい。

本システムで実装した仕組み、すなわち、周辺の Bluetooth 信号を追跡することによって、匿名の ID と行動が結び付けられたデータベースをクラウドソーシングにて作り上げる仕組みには、様々な利用方法が考えられる。今後は、電車だけでなく、食堂やタクシーの待ち行列等への応用を考え、より有効に使えるようなシチュエーションや使い方を考えていきたい。

8. 謝辞

本研究は独立行政法人用法処理推進機構の 2014 年度未踏 IT 人材発掘・育成事業に採択され、支援を受けて開発を行った。

参考文献

- [1] Maekawa, Y., Uchiyama, A., Yamaguchi, H. and Hishigashino, T.: Car-level Congestion and Position Estimation for Railway Trips Using Mobile Phones, *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, New York, NY, USA, ACM, pp. 939–950 (online), DOI: 10.1145/2632048.2636062 (2014).
- [2] Muñoz-Organero, M., Muñoz-Merino, P. J. and Delgado Kloos, C.: Using Bluetooth to Implement a Pervasive Indoor Positioning System with Minimal Requirements at the Application Level, *Mob. Inf. Syst.*, Vol. 8, No. 1, pp. 73–82 (online), DOI: 10.3233/MIS-2012-0132 (2012).
- [3] 牛越達也, 出射健一郎, 西出亮, 河野恭之: Bluetooth デバイスの検出履歴を用いたユーザ行動の分類, 技術報告 2, 関西学院大学大学院理工学研究科, 関西学院大学大学院理工学研究科, 関西学院大学大学院理工学研究科, 関西学院大学大学院理工学研究科 (2009).
- [4] 康裕田岡, 太 納谷, 春生野間, 潔 小暮, 周浩 李: Bluetooth の電波強度を用いたユーザの位置推定手法, 情報処理学会研究報告. UBI, [ユビキタスコンピューティングシステム], Vol. 19, pp. 147–152 (オンライン), 入手先 (<http://ci.nii.ac.jp/naid/110006861986/>) (2008).

^{*6} <http://www.jreast-app.jp/>

^{*7} <http://www.navitime.co.jp/?ctl=0171>