

SUWANT!:

無線IDを利用した乗客乗降駅情報の収集・提供システム

笹川 真奈^{1,a)} 椎尾 一郎^{1,b)}

概要: 電車内で乗り合わせた乗客の降りる駅を提示することによって席に座れるチャンスをユーザに提供するスマートフォンアプリケーション SUWANT!^{a)}を我々は開発した。SUWANT!は乗客が所有する携帯機器のBluetooth信号を利用して乗客の乗降駅情報を取得し、これらをクラウド上の共有データベースに多数集積し利用することによってユーザ周辺にいる電車乗客の降車駅を推定する。ユーザの周辺にすぐ降りそうな乗客がいれば、その存在とユーザからの推定距離・推定降車駅がスマートフォン上に表示される。必要ならば座っている乗客の周辺を歩き回りながら推定距離の変化を見ることによって降りそうな乗客を探し当てることもできる。本システムはクラウドソーシングに基づいているため、交通機関利用者の移動分析やマーケティングなどの様々な用途に利用可能な膨大なデータを収集することが可能である。本システムの評価実験を実際の電車内で行ったところ、特定の電車区間において10回乗車中2回満席の電車内で座ることができた。

^{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! : System for Gathering and Presenting Passengers' Getting-off Stations by Using Wireless Signals' IDs.

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

Abstract: SUWANT!^{a)} is a smartphone application using Bluetooth that helps users on a train find passengers sitting nearby who will disembark at the upcoming stations; this applications is powered by crowdsourcing. 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 twice in 10 rides on a specific train.

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

Keywords: Smartphone application; Bluetooth; Crowdsourcing; Train;

1. はじめに

近年クラウドソーシングを用いて幅広く大量のデータを

収集する方法が着目されている。そのデータ収集方法は特定の端末やフォーム等でユーザが手入力するのが主であったが、最近ではスマートフォンの普及に伴いスマートフォンに搭載された高機能センサからの検出値を用いたクラウドソーシングが可能になりつつある [2]。このようなクラウドソーシングに多数のユーザを継続的に参加させるため

¹ お茶の水女子大学
Ochanomizu University, Bunkyo-ku, Tokyo 112-8610, Japan
^{a)} g1120520@is.ocha.ac.jp
^{b)} siiio@acm.org

には参加したユーザが日常生活で恩恵を受けられるシステムが必要である。例えば電車等の公共交通機関利用者の移動情報をクラウドソーシングにより収集したい場合、車両から降りる人がわかり電車で座りやすくなるという恩恵をユーザに与えることでクラウドソーシングに参加するユーザを増やすことができるであろう。

都市部で通勤・通学をする多くの日本人にとって電車で空席を見つけることはクラウドソーシングに参加するための重要な動機付けになる。東京都では大多数の人々の通勤手段が電車であることに加えて*1、日本人の電車での平均通勤時間は1時間である*2。このように日本人の電車への依存と長い通勤時間から、電車で座席に座り前日の残業等で不十分となった睡眠を通勤時間中に補いたい等と感じる人は多いと察せられる。しかし一方で通勤電車等の混雑した電車で座席に座れるかどうかは運によるところが大きい。例えば、すぐに降りる乗客の近くに偶然立っていれば早く座ることができるが、自分よりも後に乗車してきた乗客がたまたますぐに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。

そこで我々は、電車で座って休みたいと強く願っている人々のために、混雑した電車内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のように電車内でBluetooth信号を用いて乗客の情報を集めた後に、収集し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。

2. Bluetooth による乗客行動推定

現在日本人の半数以上がスマートフォンを所持しており、ほぼ全てのスマートフォンにはBluetooth通信機能が内蔵され様々な周辺機器との接続に用いられている。ヘッドフォン、スマートリストバンドやスマートウォッチ等のBluetooth接続可能なウェアラブル機器が普及していることから、電車内でスマートフォンのBluetoothをONにする乗客は多く、今後もさらに増加すると考えられる。



図1 SUWANT!使用場面のイメージ図

*1 <http://www.stat.go.jp/data/kokusei/2000/jutsu1/00/04.htm>
 *2 <http://www.mlit.go.jp/common/001001523.pdf>

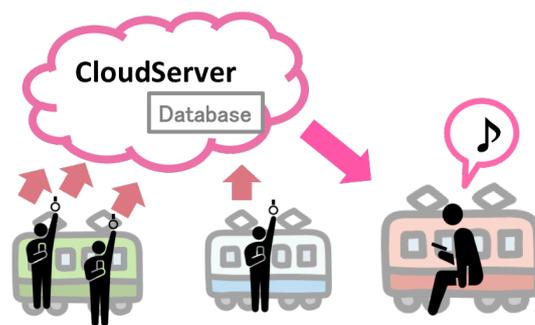


図2 SUWANT!クラウドソーシングのイメージ図

Bluetooth通信機能を持つ端末(以下Bluetooth端末とする)の半径10m程度以内にBluetooth検知可能モードになっている他のBluetooth端末があれば、その端末独自のBluetooth IDと電波強度(以下RSSIとする)を取得することができる。これらのことからBluetooth信号を用いれば乗客の行動が推定できると我々は考えた。すなわち、ある乗客の持つBluetooth端末のBluetooth IDが停車駅で検知・消失することを長期的に記録することで、その乗客の乗車・降車駅を特定できる。さらに、このデータを大量に用意した上で車両内のBluetooth IDと各々のRSSI値を追跡することによって、周辺のBluetooth端末を持つ乗客の車内位置と乗降駅を推定できると考えた。

本システムでは乗客の乗降駅情報を多く集めるほどユーザの周辺乗客が降りる駅をより正確に提示できるが、膨大なデータを一人で集めるには多大な時間がかかる。そこで我々は、本アプリケーションの多数のユーザで集めたデータを全ユーザで共有することで席に座れる確率を高めることを目指した。図2に示すように本システムでは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してクラウドソーシングの手法で乗客の乗降駅データベースを構築する。このためユーザが増えれば増えるほどすぐ降りそうな乗客を探し当てられる確率が増加することが期待できる。本アプリケーションはONにするだけで情報収集を自動で行うため、ユーザは手間なく情報収集に貢献できる。

2.1 電車内Bluetooth ID取得数の実地調査

本アプリケーションSUWANT!の仕組みはBluetooth検知可能モードの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では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BluetoothIDを12秒毎*4に取得し、タイムスタンプ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。時間帯を

*3 測定にはXperia Z1を用いた。
 *4 実装時に使用したAndroidの標準APIでは、最短で12秒毎にBluetoothIDを取得できる。

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)

5 つ (通勤/午前/午後/夕方/帰宅) に分類し、筆者のうちの一人が各々の時間帯ごとに前方・中方車両にて各車両 2 周ずつ計 20 周 JR 東日本の山手線に乗車して Bluetooth ID を取得した。1 駅間の平均取得 Bluetooth ID 数を図 3 に示す。調査の結果、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。ユーザの周辺乗客のうち最低 1 人でも Bluetooth 信号によって乗降駅が分かれば、本アプリケーション上で乗降駅情報を表示できる。そのためこの結果から実際に運行されている電車内で本アプリケーションは機能すると判断した。

2.2 RSSI 値と距離の関係

本アプリケーションは乗客が所持している Bluetooth 端末の RSSI 値を用いて、近くの停車駅で降車すると推測された乗客までの距離を推定して表示する。RSSI 値は周囲の環境の影響を大きく受けると予想されるため、RSSI 値と距離の関係を実際の電車内にて調査した*5。

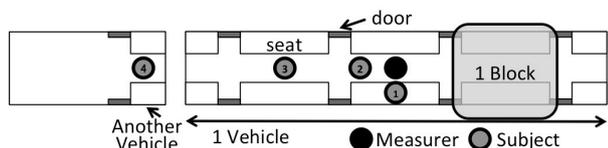


図 4 混雑度別実験での測定者・被測定者の配置図

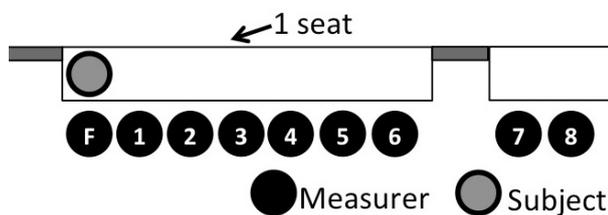


図 5 中混雑度実験での測定者・被測定者の配置図

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

*5 調査では送受信とも Xperia Z を用いた。

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

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

表 3 SUWANT!における RSSI 値と距離の関係

取得 RSSI 値	対象乗客とユーザ間の推定距離
-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

2.2.1 混雑度別：RSSI 値と距離の関係

Bluetooth 信号の RSSI 値は障害物によって変動することから車内の混雑度に依存すると予想される。そのため、表 1 に示すように混雑度を 3 段階に分類して各々の混雑状況において次の測定を行った。検知可能な 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)に存在するときに、被測定者の RSSI 値を各々 10 回ずつ測定するとことによって RSSI 値と距離の関係を調査した。その結果、同じ距離であっても混雑度によって受信する RSSI 値が表 2 に示されているように異なることがわかった。

2.2.2 中混雑度：RSSI 値と距離の関係

本アプリケーションのユーザは降車しようとする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ユーザの移動が困難なため、本システムを使用しても席の確保は難しい。逆に空席のある車両では、本システムは使用されない。よって本システムが使用される場合の混雑度は中混雑度が主だと考えられる。そこで距離間隔をさらに細かくして、中混雑度での RSSI 値と距離の関係を再調査した。調査では、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図 5 に示すように座らせた。次に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 信号の RSSI 値を測定した。図 6 に測定結果の平均値を示す。Bluetooth の電波は不安定なため、実際の測定値は平均値の上下に分散している。そこで本アプリケーション上では、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, 8 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ピクトグラムを用いて示す。

3. SUWANT!

3.1 画面説明

本アプリケーション SUWANT!は混雑した電車内において近くの駅で降りると予想される乗客をユーザ周辺で探知することにより、ユーザがより効率的に席に座ることを支援するスマートフォンアプリケーションである。本アプリケーションは周辺の乗客が所持する Bluetooth 端末の Bluetooth 信号を検出した後に、クラウドソーシング手法によって作られた降車駅データベースを用いて電車内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左下を原点とした 2 次元グラフのように見る。すなわち、横軸は対象乗客とユーザ間の推定距離を示しており、原点から右方向に進むほど推定距離は遠くなる。縦軸はユーザがいる路線図を示しており、対象乗客の推定乗降駅を表示する。そして、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。この画面の左側 (y 軸) には現在駅 (原点位置) と乗車している路線、そしてこの先の停車駅が 3 駅分表示される。つまり図 7 で左側の停車駅のリストの一番上に表示されている駅が現在駅から 3 つ先の停車駅である。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いるため、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。本システムでは端末の位置情報から路線を推定しているが、乗車している路線と並行する別路線がある場合には取り違える可能性がある。その場合ユーザは現在路線が表示されているボタンをタップして表示されるポップアップメニューから正しい現在路線を選択する

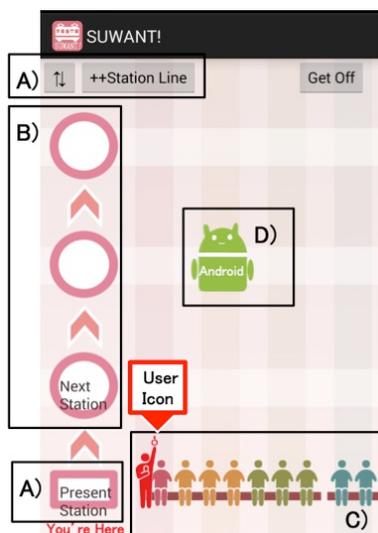


図 7 SUWANT!初期画面：A). 現在路線、駅名 B). 到着駅指標 (y 軸) C). ユーザからの距離指標 (x 軸) D). 端末機器の種類

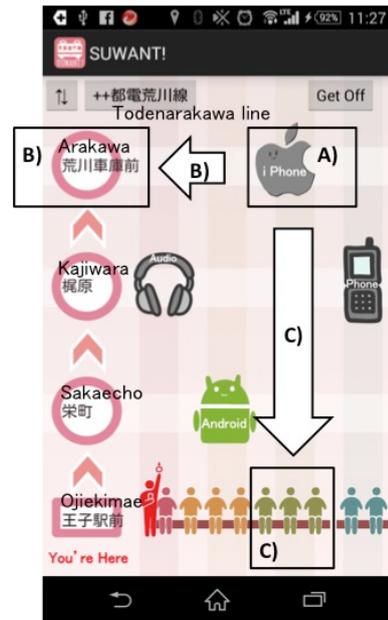


図 8 SUWANT!画面例：A). iPhone 所持 B). 荒川車庫前で降りる C). ユーザから 4-6 席分離れている

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い進行方向に変更することができる。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表している。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す。

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。そして、もしユーザの周辺 (座席 8 席分以内) に近くの停車駅で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画面に出現する。アイコンの形状は対象乗客が所持している Bluetooth 端末の機器の種類、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おり、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。画面上に示されている種類の端末を対象乗客が使用していることが明確に確認できれば、ユーザはこのアイコンにより対象乗客を特定することができる。アイコンの横方向の位置はユーザと対象乗客の距離に応じて変化するため、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、ユーザが遠のけば右に動く。またアイコンの縦方向の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。

3.2 使い方

SUWANT!の使用法を実際のアプリケーション画面のスクリーンショット (図 8) を用いて説明する。

- (1) ユーザが電車に乗車した際に SUWANT!を起動する。

(2) SUWANT!は図8に示すように周辺の Bluetooth 端末の機器（この例では4台）を検知して画面上に情報を提示する。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3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を周辺乗客の中から特定する。一つの手段としては、車内を歩き回りながら変化するアプリケーション画面を見て対象乗客を探す、ということが考えられる。電車内をユーザが歩き回れば対象乗客との距離が変わりアイコンが左右に移動するため、もし対象乗客の目の前に来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。

(4) 対象乗客を発見できれば、ユーザは対象乗客の前に立ち推定降車駅まで待つことで対象乗客が降りた席に座ることができる。

4. 実装

本アプリケーションは Android OS 4.4.2 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 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 ID とその RSSI(以下 Bluetooth 情報と呼ぶ)を取得し続ける。なお 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 端末、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。このように本システムは Bluetooth 情報を取得する一方で、取得した情報を用いて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）にてユーザ周囲にある Bluetooth 端末の情報を収集した後にサーバで乗客の乗降駅データベースを構築する。図9に本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 情報等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は検知した Bluetooth 情報と現在位置、タイムスタンプの3種類である。ここで現在位置は Android の標準 API によって取得された緯度経度を用いる。

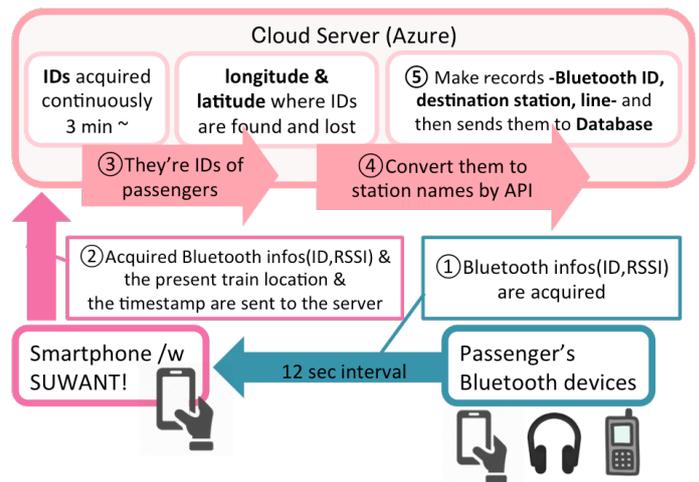


図9 システム構成図

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側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

まず生データからノイズを取り除くために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、その乗客が所有する端末の Bluetooth 信号が連続して取得できないことがある。この場合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端末の ID として処理をする。これは東京都では同じ車両に1時間以上乗る乗客は少ないと判断したためであり、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必要がある。もう一つの処理は 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 信号や、すれ違った電車あるいは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収集をしているユーザと同じ電車の車両に乗っている乗客ではないと認識し分析対象から除外する。実装では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た。ここでも東京都での駅間が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に閾値を調整する必要がある。

(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 web API^{*6} 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。これらの情報から - Bluetooth ID, 降車駅名, 路線名 - の組 (以下レコードと呼ぶ) を作りデータベースに登録する。本システムはクラウドソーシングの手法を用いているため、本アプリケーション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充実し近くの駅で降車する周辺乗客の推定がより正確になると考えられる。

4.2 情報提示

本システム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から近くの駅で降りる可能性のある乗客を推定し、クライアントアプリケーション上でユーザに提示する。周辺乗客の降車駅は Bluetooth ID から推定し、ユーザとの距離は RSSI の強度を用いて推定する。

クライアントアプリケーションが Bluetooth 信号を検知すると取得した Bluetooth ID とユーザの現路線名を合わせ持つレコードが登録されているかどうかサーバに対して問い合わせを行う。もしデータベースにそのレコードが登録してあれば、サーバはそのレコードの降車駅名をクライアントに返す。その際に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。現在の実装では、複数の降車駅名がデータベースに登録されていた場合、クライアントユーザが乗車している路線にある全ての降車駅名を返している。将来的には、対象乗客の駅ごとの降車頻度・時間帯・曜日等によってフィルタリングをし、より有用な降車駅をユーザに返すようにしたいと考えている。クライアントはサーバから降車駅を受け取ると、その Bluetooth 端末所持者までの距離を RSSI 値を利用して推定する。ユーザからの推定距離は第 2.2 節で述べたように RSSI 値を用いて表 3 に基づいて表示する。しかしながら、関連研究 [4] にも示されている通り RSSI 値だけで正確に距離を推定するのは困難であるため、ここで提示する距離情報はあくまで目安である。また、この手法では対象乗客のいる方向を知ることは難しいが、ユーザが車両内で移動して画面に表示される距離の変化を見ることで推定可能である。

5. 評価実験

本アプリケーション SUWANT! をレコードの取得効率と

ユーザが席に座れた成功率の二つの面から評価する。

5.1 レコードの取得効率

レコードの情報を取得する効率を測るために 20 人のユーザ^{*7}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。その結果を表 4 に示す。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、全ユーザが集めたレコード数の合計は 1675 組であった。各ユーザが往復で 2 回電車に乗ることを考えると、平均して 1 回の乗車あたり約 2 組の新しいレコードが得られることがわかる。

表 4 20 人のユーザが集めたレコード数

ユーザ番号	使用日数	取得組数	1 日の平均取得レコード数 (1 回乗車の平均取得レコード数)
1	25	77	3.1(1.5)
2	9	9	1.0(0.5)
3	13	129	9.9(5.0)
4	7	3	0.4(0.2)
5	15	6	0.4(0.2)
6	47	473	10.1(5.0)
7	4	12	3.0(1.5)
8	16	44	2.8(1.4)
9	67	382	5.7(2.9)
10	13	150	11.5(5.8)
11	21	64	3(1.5)
12	48	289	6.0(3.0)
13	6	11	1.8(0.9)
14	2	13	6.5(3.3)
15	40	187	4.7(2.3)
16	10	44	4.4(2.2)
17	6	32	5.3(2.7)
18	25	39	1.6(0.8)
19	13	53	4.1(2.0)
20	13	150	11.5(5.8)
合計数	392	1675	4.3(2.1)

5.2 成功率

本アプリケーションを使用した際にユーザが席に座れる確率が上がるのかを実際の電車内で検証した。検証は筆者のうち 1 人が表 5 で示す条件下において行った。本検証を行った区間は、本システムを用いた情報収集のみを行った際に 10 回の乗車中 1 回も座ることのできなかつた区間である。これらの電車の混雑度は、席は埋まっているが適度に車内を歩き回れる程度であった。しかし、いずれも都心部へ向かう路線である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な状況であった。さらに駅間の距離が他の路線よりもやや長いため、より座れるチャンスが

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

*7 全員が自身の android 端末を所持している

表 5 検証時の乗車電車条件

路線名	出発時刻	区間	停車駅数 (列車種)
京王線	2015/2/16-20 8:52	仙川から新宿	11(各駅)
中央線	2015/2/16-20 10:16	立川から新宿	15(快速)

少ないことから、本検証を行った区間は本アプリケーションが想定する状況として適していたと考えられる。また本検証を行う前に、検証を行う区間内で降車すると推定されている乗客のレコードを事前に各区間 100 組以上データベースに収集した。これらはユニークな ID のレコードである。京王線では筆者含むユーザ 2 人が 52 日間^{*8}の情報収集を行い 257 組のレコードを収集した。片方のユーザは通勤時に使用し 6 時-7 時と 17-18 時に主に収集した。もう一方のユーザ (筆者) は通学時に使用したため特定の時間帯に限らず収集を行った。中央線ではユーザ 2 人が 21 日間^{*9}の情報収集を行い 127 組のレコードを収集した。双方のユーザは通学時に使用したために特定の時間帯に限らない収集を行った。

実験結果を以下に述べる。10 回の検証中 6 回は、乗車中に車両を移動しデータベース上に既に推定降車駅が登録されている BluetoothID を探したが見つけられなかった。残りの 4 回では各々、本システムによって周辺乗客の中の 1 人の推定降車駅が表示された。そのうち 1 回は対象乗客を特定する前に乗客が降り表示が消えてしまったため特定することができなかったが、残りの 3 回では各回 1 人の対象乗客を特定することができた。特定した 3 人のうち 1 人は推定降車駅で降りずに終点まで乗っていたため筆者は座ることができなかった。別の 1 人は表示された通りの駅で降車したため座ることができた。さらに残りの 1 人は予想していた駅よりも早い段階で降りたが、降りる前にその乗客の前に立つことができたため座ることができた。この乗客は本来降車する駅よりも前の駅で快速から中央特快に乗り換えたと考えられる。以上のように SUWANT! を用いて実験を行った結果、座れる確率が 0% であった区間において 10 回の乗車中 2 回座ることに成功した。座れた時には使用区間のレコードが 100 組以上あった。第 5.1 節で述べたように 1 回乗車すると平均して 2 組の新しいレコードを得ることができるため、50 人以上のユーザが理想的な状況下で情報収集をすれば 1 日で 100 組以上のレコード集めることができる可能性がある。理想的な状況とは、各々のユーザが別の車両で情報収集を行い、かつ周辺にデータベースに登録されていない新たな BluetoothID が 2 つ以上検知される状況である。実際には車両の混雑状況や乗り合わせている乗客等の収集率を左右する様々な要因があるため、100 組以上のレコードを集めるために必要なユーザ数や日数は、想定しているよりも多く必要になると考える。

^{*8} 2014 年 9 月 2 日から 2015 年 2 月 19 日の間のうちの 52 日間

^{*9} 2014 年 9 月 5 日から 2015 年 2 月 17 日の間のうちの 21 日間

6. 関連研究

本研究以外にもクラウドソーシングを基にして匿名のスマートフォンのセンサ値を用いて様々な情報を収集する研究は多数実施されている。水上ら [3] は、iBeacon^{*10} を設置し、この周辺にあるスマートフォンのセンサを用いて温度等の iBeacon 周辺の環境情報を得るクラウドソーシングシステムを実装した。

また Bluetooth 信号を用いて人の行動や位置を推定する研究も [1] で示されているように多数存在する。牛越ら [5] は外部環境中にある固定局等からの Bluetooth 信号の検知パターンによってユーザの行動を推定した。田岡ら [6] 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。また前川ら [4] は電車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。

次に、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。“電車で座る”^{*11} は特定の駅で降りた人の特徴を手入力でユーザが記録するアプリである。これは次の駅で降りる人を見つけ早く席に座りたいという目的のために乗客情報と降車駅を結びつける点は本システムと同じである。しかし本システムと比較するとデータベース入力がユーザ個人の手入力であるために手間がかかり、さらにユーザ間でデータ共有をしないために得られるデータ量が限られると考えられる。“トレインネット”^{*12} 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り、このアプリケーションを乗車前に活用することで比較的空いている車両に乗ることができる。また“こみれぼ”^{*13} 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ユーザはより空いている電車や路線を選択できる。これら 2 つのアプリケーションはユーザがより空いている車両に乗車することを支援するため、ユーザは混雑した電車内で席に座れる確率を高めることができる。しかしこれらは車両単位での電車の混雑度を示すため、本システムのように乗車中にこれから降車する乗客を特定することはできない。

7. まとめと今後の課題

本研究では公共交通機関で効率よく座れることを支援するスマートフォンアプリケーション SUWANT! を開発した。本アプリケーションは、クラウドソーシングにより集積した乗客の Bluetooth 情報を用いてもうすぐ降りる乗客

^{*10} <https://support.apple.com/ja-jp/HT202880>

^{*11} <http://applion.jp/iphone/app/447550582/>

^{*12} <http://www.jreast-app.jp/>

^{*13} <http://www.navitime.co.jp/?ctl=0171>

を見つげ出し、その乗客情報をユーザに知らせる。開発したアプリケーションを東京都の特定の電車区間で使用した結果、100組以上その区間の乗降駅データがある状況下において、10回乗車中2回満席の電車内で座ることができた。これは本アプリケーションの有用性を示すだけでなく、乗客の乗降駅情報をクラウドソーシングにより収集するシステムに対しての十分な動機付けになることを示している。そのため本システムを用いて交通機関利用者の移動分析やマーケティングなどの様々な用途に利用可能な膨大なデータを収集できる可能性がある。

今後は積極的にユーザを増やすことによってデータベースに蓄積するデータ量を増やすことを目指す。そのためにAndroidアプリケーションだけでなくiOSアプリケーションの開発も進めると共に、サーバの容量を増加することを計画している。さらに乗客行動推定の精度を上げるために生データをより有効に分析するアルゴリズムの開発も検討している。

現在の実装の一つの問題点として、電車の乗降の際にユーザが手でアプリケーションを起動・終了する必要があることが挙げられる。実際、評価実験のユーザの中には電車に乗車したにもかかわらずアプリケーションを起動し忘れデータを取り損ねるユーザがいた。そこで、この点を改善するために樋口ら [7] の論文にあるような電車の発車・停車を自動的に判定する仕組みを取り入れることを検討している。電車の乗り降りの際に自動的にデータ収集を開始・終了するような実装を行いユーザの手間を省くことで、確実にデータを集められるようなシステムに改善したい。

将来的には実サービスとして本研究を提供したいと考えているが、そのためには様々な問題が想定される。一つは実環境下で取得できるBluetooth信号数の限界である。今回行った探索で見つけたBluetooth IDの端末は、フィーチャーフォン50%、スマートフォン30%、その他20%程度の割合であった。フィーチャーフォンはデフォルトでBluetooth検出が可能な設定にされている傾向にあるため、比較的高い割合で発見できたと推察される。スマートフォンについては、必要な場合のみ検出可能設定されることがあり、実際の普及率に比べて比較的低い割合で発見されたと考えられる。そこでBluetooth信号だけでなくWi-Fiの信号を併用することで、情報収集対象の乗客を増やしたい。例えば、ビーコン^{*14}を利用することによって、モバイルWi-Fiルータやテザリング中の端末を所持している乗客の情報を取得したい。さらに、Probe Request^{*15}を利用することによって、電車内でWi-Fiを利用している乗客も情報収集対象にしたい。もう一つの問題点は、ユーザが増加するに従って、本システムを使用した席取りに参加する人が増え、ユーザ間の競争が激しくなることが挙げられる。そ

れを解決するために、今後はユーザ内で優先順位を付ける機能を実装していきたい。一例として、先に電車に乗ったユーザや情報収集の貢献度の高いユーザに優先的に情報を開示したり、優先度の高いユーザが狙っている乗客情報を他のユーザから見えないようにしたりする、といった機能が考えられる。

今後はユーザ数を増やし再検証を行いたい。それと並行して、ユーザのフィードバックを基に考察を行い、議論・検証を通して研究を深めたい。

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] Mario, M.-O., Pedro, J. M.-M. and Carlos, D. K.: Using Bluetooth to Implement a Pervasive Indoor Positioning System with Minimal Requirements at the Application Level, *Mobile Information Systems*, Vol. 8, No. 1, pp. 73-82 (online), DOI: 10.3233/MIS-2012-0132 (2012).
- [2] Prashanth, M., Venkata, N. P. and Ramachandran, R.: Nericell: Rich Monitoring of Road and Traffic Conditions Using Mobile Smartphones, *Proceedings of the 6th ACM conference on Embedded network sensor systems*, SenSys '08, New York, NY, USA, ACM, pp. 323-336 (online), DOI: 10.1145/1460412.1460444 (2008).
- [3] Takamasa, M., Katsuhiko, N., Chiaki, D., Tomohiro, N., Ken, O., Hiroshi, I., Takaaki, H. and Tadanori, M.: Fundamental Design for a Beacon Device Based Unconscious Participatory Sensing System, *Proceedings of the International MultiConference of Engineers and Computer Scientists*, IMECS 2015, Vol. 2, pp. 593-598 (2015).
- [4] Yuki, M., Akira, U., Hirozumi, Y. and Teruo, H.: 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).
- [5] 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, 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. 2009, No. 2, pp. 1-8 (オンライン), 入手先 (<http://ci.nii.ac.jp/naid/110007995115/>) (2009).
- [6] 田岡康裕, 納谷太, 野間春生, 小暮潔, 李周浩: 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).
- [7] 樋口雄大, 山口弘純, 東野輝夫: スマートフォン内蔵センサを用いた鉄道乗車コンテキストの推定, 情報処理学会研究報告. MBL, [モバイルコンピューティングとユビキタス通信], Vol. 35, pp. 1-8 (オンライン), 入手先 (<http://id.nii.ac.jp/1001/00142140/>) (2015).

*14 Wi-Fiのアクセスポイントが発している信号

*15 Wi-Fiのアクセスポイントを探索する際に端末が出す信号