

個人セキュリティ GPS の開発

奈良部凌也^{†1} 綿鍋明日菜^{†1} 森信一郎^{†1}

概要: 現在, スマートフォンの位置情報からランドマークを推定した SNS サービスが普及している. 位置情報取得にはユーザー情報をサーバーへ送信する必要があるがその際, ユーザーの情報が不正に取得される恐れがありプライバシーの安全性が問題視されている. サーバーを使わない位置推定技術の提案も行われているが推定時間が長い, 消費電力が大きい, 専用のインフラが必要など課題も多い. 原因の一つに位置推定に必要な計算量が大きく, スマートフォンでの実装に適していない事があげられる. そこで本研究ではスマートフォンにおけるランドマークの推定において最も計算負荷の小さなアルゴリズムの検討を行った. 検討したアルゴリズムを端末へ実装し評価を行ったので報告する.

キーワード: WiFi, 屋内測位, 機械学習, デバイス

Development of personal security GPS

RYOYA NARABE^{†1} ASUNA WATANABE^{†1} SHINICHIRO MORI^{†1}

Abstract:

Currently, SNS services that estimate landmarks from smartphone location information are in widespread use. In order to obtain location information, it is necessary to transmit user information to the server, but at that time, there is a possibility that the user's information may be acquired illegally, and privacy security is regarded as a problem. There have also been proposals for location estimation techniques that do not use servers, but there are many issues such as long estimation times, high power consumption, and the need for dedicated infrastructure. One of the causes is that the amount of calculation required for position estimation is large and it is not suitable for implementation on a smartphone. So, in this research, we considered the algorithm with the least calculation load in the estimation of the landmark in the smartphone. As we implemented the evaluated algorithm on the terminal and evaluated it, we report it.

Keywords: WiFi, indoor positioning, machine learning, device

1. 背景

近年コンシューマー分野で急速に普及したスマートフォンやタブレットなどの携帯端末を活用した「コンシューマライゼーション」が加速している. コンシューマライゼーションとして拡大しているサービスの一つに位置情報を活用した LBS(Location-Based Service)がある. LBS はモバイル端末で取得した位置情報を利用してユーザーに役立つ情報を提供するサービスであり, 欧米を始め日本でも急速に拡大している. 一方, 第三者に位置情報を利用される事件も増加している. 他人の行動を追跡するストーカー行為や, 他人の留守を確認した盗みなど社会問題として増加しつつある. 個人の位置情報は公共の回線を利用しないようにすべきであるが, 現状の位置情報サービスはクラウドサーバで利用者の位置を計算しているものが主流である. サービス利用時にユーザーの位置情報を推定するための環境情報をサーバーへ送信する. サーバーはその情報を元にユーザー位置を推定してユーザーに返信する. 従って, ユーザーの位置情報や移動履歴情報がインターネット上に保存及び

送受信される事となる. 情報が漏洩すると悪意ある第三者がユーザーのプライバシーを侵害する可能性が発生する. このような個人の位置に関するプライバシーは位置プライバシー (Location Privacy) と呼ばれ, Duckham と Kulik によれば「自分の位置情報をいつ, どのように, どの程度他人に知らせるのかを自分自身で決定する権利に関連する, 特定の種類の情報プライバシー」[1]であるとしている.

また, LBS の位置情報の表現方法も変化してきている. これまではナビゲーションで使われている緯度経度情報が主流であったが, 最近はランドマークによる表現が増加している. 中澤, 北らは「ランドマークは歩行者が自身の位置と地図上の位置とを対応付けるために用いられ, 移動する際の重要な手がかりとなる」としている. [2]人間にとっては緯度経度情報より, ランドマーク情報の方が場所を直感的に理解できるため利便性が高い. 最近 SNS で利用されている CheckIn/CheckOut の概念はランドマークを使った位置表現方法となっており, 参照する利用者側も大変わかりやすい. 遊園地などのレジャーランドにおいてもランドマークの利用は始まっており, 今後はナビゲーションなどを除いて多くの LBS は位置表現としてランドマークが採用されていくと思われる.

^{†1} 千葉工業大学
Chiba institute of technology, Japan

以上のことからランドマーク単位で位置が推定でき、かつ個人の位置情報をクラウドで扱わない仕組みが必要となる。基本的に個人の位置情報を漏らさないようにするためには、端末だけで位置を推定する仕組みを取り入れればよい。しかしながらそのような仕組みが未だ使われていないのは、位置推定に必要な処理量が携帯端末で推奨される処理量を超えている点にある。また、これまでもGPSで測位した情報に誤差を加算してユーザー位置の推定を困難にする方法も検討されてきた。がGPSはその仕組み上、常に使うには消費電力が大きすぎる。GPSの代わりにWiFiなどの情報を事前に取得して機械学習を行う方法も提案されているが、端末のみで処理をすることが難しい。これは多くの機械学習アルゴリズムは学習データが増加すると指数的に処理量が増加する点にある。教師データとなる環境情報は常に変化するため、環境情報の更新は必要不可欠である。その為、使うごとに学習すべきデータが増加し、処理する端末の処理量が増加する悪循環に陥ることになる。

本論においては、「ランドマーク単位で自分の位置を認識」でき、「極力軽量の計算量」でかつ「学習データが増加しても処理量が増加しない」方法について論じる。2章では関連研究について述べ、3章では上記の課題を解決する提案手法を述べる。4章では提案手法を実際に端末に実装して評価を行い、6章でまとめを述べる。

2. 関連技術

本章では位置測位に関する関連技術について述べる。2.1節、2.2節ではサーバーを使わないiBeacon、RFIDを用いた位置測位技術について、2.3節ではWiFiを用いた位置測位技術についてそれぞれ述べる。

2.1 iBeacon

2013年6月に米国で行われた「WWDC2013」で発表されたiBeacon[3]は、米国Apple社のiOS7から標準搭載されたBluetooth Low Energy(BLE)を使った近距離無線通信技術のことである。発信側のiBeaconから発信される情報を利用し、それに対応した受信側のスマホ端末と連動させることで設置店舗の情報や場所が分かる。Android端末でもAndroid4.3からBLEに対応したチップが搭載され利用できるようになった。デバイスがiBeacon端末に近づいたときにはアプリが起動していなくてもユーザーへプッシュ情報を配信することができる。レストラン、野球場や空港など、様々な場所で活用されている。

BLEの電波は数センチから数十メートルまでの到達範囲で「近接」、「近い」、「遠い」という3種類の識別端子で位置情報を提供する。しかし、ID偽装や信号の書換えなどセキュリティ上の課題の発生や、ビーコンなどの専用のインフラを設置する必要があるため、個人での運用が難しい。

2.2 RFID

RFID(英: radio frequency identifier)とは、ID情報を埋め込

んだRFタグから、電磁界や電波などを用いた近距離(周波数帯によって数cm~数m)の無線通信によって情報をやりとりするもの、および技術全般を指す。従来のRFタグは複数の電子素子が搭載された回路基板で構成されていたが、近年小さなワンチップのIC(集積回路)で実現できるようになった。非接触ICカードもRFIDと同様の技術を用いており、広義のRFIDの一種に含まれる。日本ではFeliCa規格が支配的である。狭義では、タグとリーダとの間の無線通信技術であるが、技術分野としてはそれにとどまらず、タグを様々な物や人に取り付け、それらの位置や動きをリアルタイムで把握するという運用システム全般まで含めて語られる。

RFIDタグによる位置推定は信号を送受信することでIDを取得し、絶対的な位置情報を取得できる。この技術も先述のiBeacon同様、ビーコンなどの専用のインフラを設置する必要がある、個人での運用が難しくなるという問題がある。

2.3 WiFiでの位置測位

GPSを携帯して街中を移動しながら得た大量の無線LAN電界情報のログから、アクセスポイントの位置を推定し、エンドユーザによるアクセスによってアクセス場所の位置の推定や、位置情報データベースの更新を特徴とする無線LAN位置情報システムがある。[4]このシステムの特徴は、WiFiセンシングによって得た電測情報を位置情報(緯度経度・住所)に変換するWebサービスを提供することや、フロア情報(1F, 2F, 構内・地下)などの情報も提供可能であることである。位置情報データベースは、GPSや地図マッピングなどを利用して位置情報が添付された電測ログ情報をもとに、アクセスポイントの位置を推定する初期フェーズと、利用者アクセスによる動的なデータベース更新のフェーズを併用する。[5]

利用者によるデータベース更新は、利用者が電測情報の測定と同時に明示的に位置を登録する場合に行われるが、利用者が単に位置を検索した場合にも発生する。つまり大量の利用者が継続的にシステムに対して検索要求を発生する行為自体が、データベースを更新・増大させるための重要な情報源となる。また、WebサービスとしてのAPIを提供するため、位置情報を利用したWebアプリケーションを容易に構築することができる。しかしながらデータベースに基づき位置情報を推定するため、自らの位置情報をサーバーにあげる必要がある、前述したプライバシー上の問題が生じてしまう。

3. 提案手法

本研究では携帯端末のみでWiFiデータを取得、学習し、新たなデータを取得した時に学習データと比較し結果を出力するというシステムを提案する。本章では本学を実験場所として、部屋識別のためのアルゴリズムの比較検討と使

用する WiFi のスキャン時間に対する検討を行った。

3.1 部屋識別のための最適性調査実験

WiFi から取得できるデータには RSSI(Received Signal Strength Indicator)がある。RSSI とはアクセスポイントまたはルータからの信号を端末が受信できる電波強度のレベルを示したものである。RSSI の分布と取得データを用いてどの程度部屋を区別出来るのかを調べるために、基準となる部屋を決め、隣接する部屋、一部屋離れた部屋、階違いの部屋の中のデータを取得した。取得に際しては部屋の中を網羅的に歩いて RSSI を連続的に測定する。測定した電波強度から部屋の識別性能について分析した。今回の実験では図 1 に示す通り、基準となる部屋 611、一つ離れた部屋 612、さらに一つ離れた部屋 613、階違いの部屋 621 の 4 部屋で実験を行った。また、星印のついていない部分に無線 LAN のアクセスポイント(AP)が設置されている。

対象の部屋の中をそれぞれ満遍なく歩き、「AirMac ユーティリティ[6]」によって観測される電波強度を測定した。その後観測された MAC アドレスについて測定した RSSI で主成分分析を行い第 3 主成分まで縮約し、可視化して分析を行った。

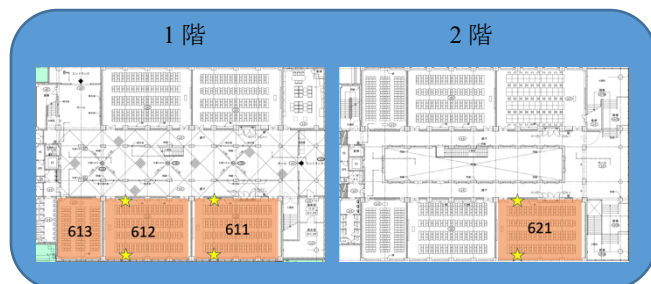


図 1 実験を行った対象の部屋
(千葉工業大学津田沼キャンパス 6 号館)

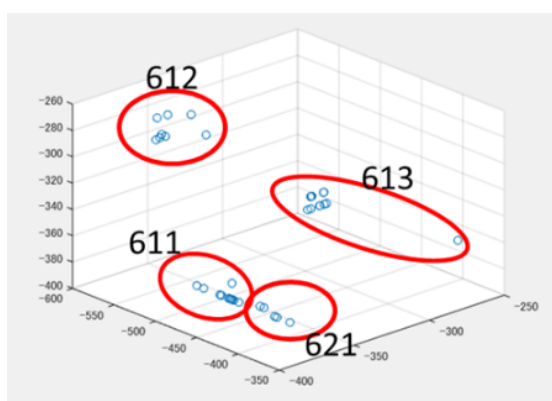


図 2 4 部屋での主成分分析分布

図 2 のグラフは 611, 612, 613, 621 の各教室で観測された電波強度を第 1～第 3 主成分上に示したものである。この結果から、各教室のデータ群が十分に離れており、厳密に最適化処理を行わなくとも各部屋の場所の区別がで

ることが分かった。またこの実験結果から、各部屋を単一の MAC アドレスで識別ができないか調査を行った。

表 1 各部屋の最も電波強度の高い MAC アドレス

部屋	MACアドレス	電波強度	部屋	MACアドレス	電波強度
611	3c:94:d5:78:30:c2	-44	634	3c:94:d5:77:f8:c2	-43
612	3c:94:d5:77:80:05	-50	635	3c:94:d5:77:f8:c4	-48
613	3c:94:d5:77:80:02	-49	636	3c:94:d5:77:f8:c0	-48
614	3c:94:d5:77:fc:c2	-50	642	3c:94:d5:7c:3e:c3	-54
615	3c:94:d5:78:09:c4	-51	643	3c:94:d5:78:c1:02	-54
621	3c:94:d5:77:d4:44	-46	644	3c:94:d5:77:71:c4	-52
622	3c:94:d5:77:c6:82	-55	645	3c:94:d5:77:71:c2	-48
623	3c:94:d5:78:0b:04	-42	646	3c:94:d5:77:71:c2	-48
624	28:8a:1c:fc:65:04	-48	647	3c:94:d5:7c:3e:c5	-52
625	3c:94:d5:77:c6:80	-56	652	3c:94:d5:7b:76:c0	-53
626	28:8a:1c:fc:65:04	-48	653	3c:94:d5:7b:c4:c2	-49
631	3c:94:d5:77:d7:05	-48	654	3c:94:d5:7b:e4:80	-53
632	3c:94:d5:77:58:42	-56	655	3c:94:d5:7b:86:45	-49
633	3c:94:d5:77:71:c2	-45	656	3c:94:d5:7b:86:42	-50

表 2 同じ MAC アドレスが取得される確率

部屋	612	621	622	623	632
取得された回数(回)	632	448	302	338	245
最も多く取得された MAC アドレスの数(個)	180	119	65	112	67

表 1 は今回実験した千葉工業大学津田沼キャンパス 6 号館の各部屋で、取得した RSSI が一番高い MAC アドレスおよびその観測値である。同色部分 (624 と 626 および 645 と 646) が同じ MAC アドレスを受信していることを示している。このことから単一の MAC アドレスで部屋識別するには難しいということがわかる。また、表 2 は各部屋で取得することができたデータ数と、データの中で最も多く取得された MAC アドレスの数を集計したものである。このことから、データ分布の中で最も多く取得できる MAC アドレスは全体の 20~30%程度しか占めていないため、最も多く観測された MAC アドレスから部屋を識別することも難しいと考えられる。

3.2 比較対象アルゴリズム

提案する手法は端末で閉じた処理を目的としているため、多量の WiFi のスキャンデータをできるだけ早く計算し、かつ処理に負荷があまりかからないようにしなければならない。そこで、部屋識別に対し k 近傍法、サポートベクターマシン(SVM)、重心比較法、尤度比較法の各アルゴリズムの処理量の比較調査を行った。

3.2.1 k 近傍法

k 近傍法とは、あるデータの分類を近傍のデータ分布の投票によって決定する方法である。各部屋で取得した AP のデータ分布を一つのグループとしてベクトル空間上にプロットし学習データとする。その後同じ空間上に観測データをプロットし、観測データと距離が最も短い学習データを求める。

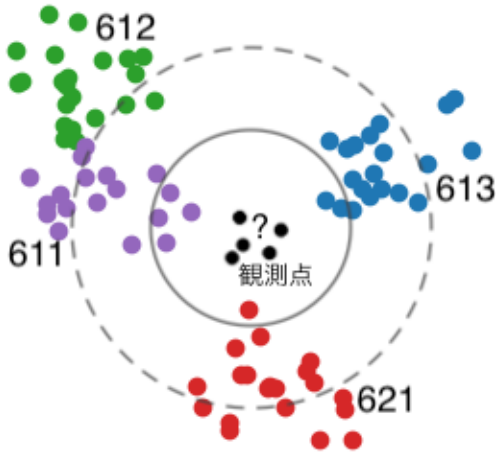


図3 k近傍法による部屋識別

図3に示すように、観測データから各学習データそれぞれに対する距離を計算し、ある範囲内で最も多く学習データが存在する学習データ群を判定結果とした。

3.2.2 サポートベクターマシン(SVM)

各部屋の観測値は十分に離れているので、ハードマージン SVM を採用した。各部屋で取得したデータ分布をそれぞれ学習させ、観測データを入力した際にどのデータ分布に属すると考えられるかを分析し、推定された部屋を分析結果として出力する。

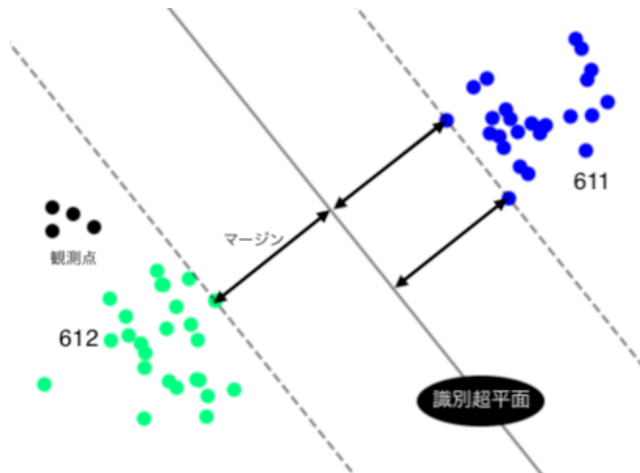


図4 SVM の分離超平面とマージン

図4の緑と青の点は各部屋のデータを表しており、点線状の点はサポートベクターを指している。点線の式は

$$a_1x_1 + a_2x_2 + b = \pm 1$$

で表し、識別超平面の計算式は、

$$a_1x_1 + a_2x_2 + b = 0$$

とする。係数の a_1, a_2, b を求めるためには

$$\max_{(a_1, a_2, b)} \left\{ \min \left\{ \frac{y_i(a_1x_{i1} + a_2x_{i2} + b)}{\sqrt{a_1^2 + a_2^2}} : i = 1, \dots, l \right\} \right\}$$

$$\text{ただし, } \{y_i(a_1x_{i1} + a_2x_{i2} + b) \geq 1, i = 1, \dots, l\} \dots \textcircled{1}$$

の最適解を求めればよい。SVM は学習過程において、すべての学習データからマージン最大化超平面を求めるため、学習データが増えると指数的に学習する計算量が増大し、処理時間も増える。また、データを追加する際には全てのデータを用いて再学習が必要となるため、さらに学習時間が増大する。観測データを入力した際に、計算しておいた識別超平面に従ってどちらのデータ群に属するのかどうかを判別し、分析結果を出力する。

3.2.3 重心比較法

重心比較法とは各学習データ分布の重心から観測データまでの距離を比較し、一番距離の短いものを結果として出力する手法である。この手法は再学習の際に必要なデータが、各 AP の累積平均と累積取得回数、および新しい取得データのみであるため、どれだけデータが増えても学習のための計算量が変わらない。そのため端末への負荷が少ないことが利点である。

学習データと取得された観測データをベクトル空間上にプロットし、学習データの重心を学習させる。あるデータ分布の重心を GSK とすると、

$$GSK = \frac{\sum_{k=1}^n rssi(k)}{n} \dots \textcircled{2}$$

と表すことができる。ここで n は取得したデータ数、 $rssi(k)$ は取得された各 MAC アドレスの電波強度を示す。取得された観測データの重心を S とすると学習データと観測データの重心のユークリッド距離 L は以下のように表せる

$$L = \sqrt{(GSK)^2 - (S)^2} \dots \textcircled{3}$$

各クラスに対してこの計算を行い、ユークリッド距離が最小となったデータ分布の部屋が観測データの取得された部屋と推定する。

3.2.4 尤度比較法

尤度比較法とは学習データと取得された観測データとの尤度を計算し、最も尤度の高いものを結果として出力する手法である。まず観測された WiFi の RSSI を S として、次のように表す。

$$S = [rssi(m_0), rssi(m_1), \dots, rssi(m_k)] \dots \textcircled{4}$$

m_0 は各 AP の MAC アドレスを表す。そして、 S の要素が観測された部屋 $C(s)$ の集合を次のように表す。

$$C(s) = \{c_0, c_1, \dots, c_m, \dots, c_l\} \dots \textcircled{5}$$

とすると、 S が観測されたときその部屋が C_m である尤度はベイズの定理に基づいて次のように表せる。

$$P(C_m|S) = \frac{P(S|C_m) \cdot P(C_m)}{P(S)} \quad \dots \textcircled{6}$$

取得データに対して尤度が一番高い部屋を推定結果として出力する。

3.3 比較試験用データの作成

3.2 で挙げたアルゴリズムの処理量における比較実験にあたり、実験対象エリアの電波状況を調査した。調査は千葉工業大学 6 号館の 621 教室と 622 教室で行った。図 3 は 621 教室で取得することができたある MAC アドレスの RSSI を横軸、度を縦軸に表したものである。このグラフを見て分かるように一つの MAC アドレスに対する RSSI は正規分布を示している。622 教室の取得データに関しても同様の結果であった。したがって今後観測されるデータもこれに準ずると考えられるため、対象教室の事前観測データにおける平均と分散を用いて試験データを生成することができる。

今回は 1 部屋あたり 400 個データを 100 種類用意し、アルゴリズムの処理量における比較実験を行なった。

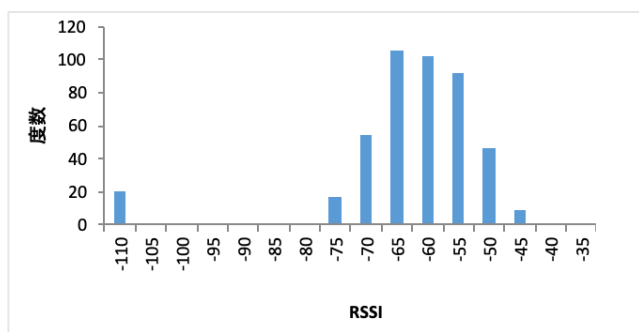


図 5 621 教室で取得された RSSI のヒストグラム

3.4 比較実験結果

100 回分のデータを用いた比較結果を以下表に示す。

表 3 誤答率と平均処理時間における識別方法比較実験

	尤度	重心	k 近傍	SVM
誤答率 (%)	15	2	0	0
平均処理時間 (ms)	93.162	29.782	94.378	46.415

表 3 において誤答率は部屋識別の結果が間違っていた数と、部屋識別の結果を得ることができなかった数を示したものである。平均処理時間はそれぞれのアルゴリズムの結果が得られるまでに要した時間を記載している。結果 k 近傍法と SVM の誤答率が一番低く、尤度比較法が一番高くなった。また、平均処理時間が一番短い結果になったのは重心比較法で、長くなったのは尤度比較法と k 近傍法という結果になった。

このことから、端末で閉じた処理をする最適なアルゴリズム法は重心比較法と SVM であると考えられる。

3.5 次元数を増やした場合の比較

重心比較法と SVM において、どちらの比較アルゴリズムが優れているか分析をするために、対象 AP を増やし、学習する時間も加味して平均処理時間の実験を行った。

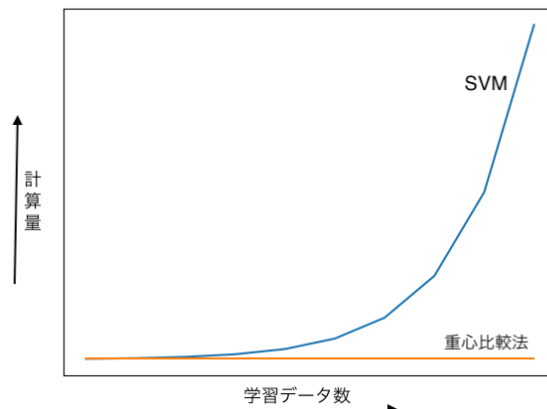


図 6 重心比較法と SVM の計算量例

表 4 重心比較法と SVM の平均処理時間の比較

	重心	SVM
平均処理時間(ms)	49.406	398.232

図 6 は、重心比較法と SVM の計算量の変化の例を表したものである。横軸は学習データ数であり、縦軸はそれによる計算量の変化を表している。重心比較法はデータ増加による再学習をしたとしても計算量は変わらないが、SVM はデータ数が増加すると全データを用いて再学習する必要があるため、計算量が指数的に増える。結果として、表 4 に示すように対象となる AP が増加し、学習する時間も処理時間に加えた平均処理時間は、重心比較法の方が短いという結果になった。従って、提案手法には重心比較法を採用する。

3.6 WiFi スキャン時間の検討

WiFi の RSSI を取得するスキャンを端末で行う時間について検討を行った。

表 5 取得時間における RSSI の平均と分散

取得時間	5秒			60秒		
	取得回数	平均値	分散	取得回数	平均値	分散
MACアドレス						
10:66:82:6f:07:fa	20	-66	0	221	-69.06	1.7558

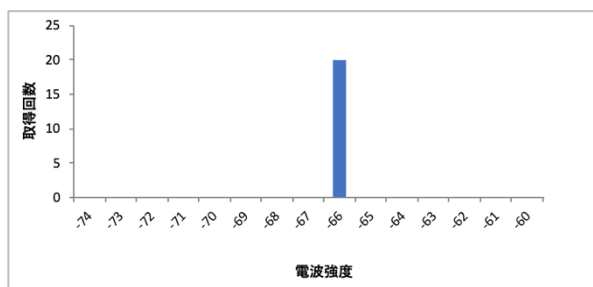


図7 取得時間が5秒の場合のデータ頻度分布

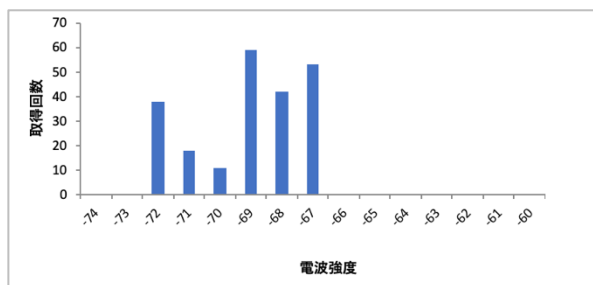


図8 取得時間が60秒の場合のデータ頻度分布

表5はある部屋で取得可能な単一のMACアドレスに対し、取得時間が5秒、60秒の場合取得できるデータの個数とRSSIの平均、分散である。図7、8はそれに対するヒストグラムである。分散の最大値は1.75であり、ヒストグラムからもわかるように、電波強度の平均値や分散は取得時間や取得回数に関わらず数dBm以内である事がわかる。つまり学習に際しては長い時間学習データを取得する必要はなく、5秒程度で十分であると言える。

4. 実験

3章よりアルゴリズムは重心比較法、スキャンの時間は5秒でよいことがわかったので、アルゴリズムを携帯端末に実装し、位置推定精度と実時間処理が可能かどうか実験を行なった。

4.1 実験環境

実験環境として、千葉工業大学津田沼校舎6号館を採用した。今回の実装プログラムのスタック図は図9のようになっている。

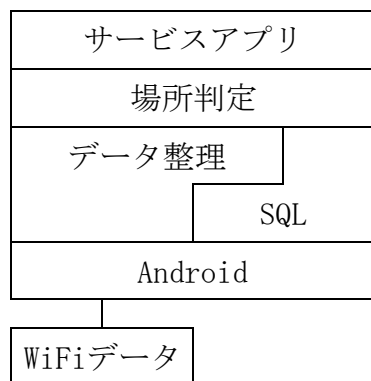


図9 実装プログラムのスタック図

WiFiのデータを取得し、学習データをAndroid端末上のSQLデータベースに格納する。場所判定の際にはWiFiのデータを取得し、取得したデータとSQLから引き出す学習データを用いて判別する。そしてその出力をサービスアプリ上で行う。

4.2 場所モデルのデータ構造

場所モデルのデータ構造を表6に示す。表6は学習したデータを格納するためのデータベース構造を表している。各部屋にIDを振り、学習した回数をカウントする。またその部屋で取得できたMACアドレスおよびRSSIの平均値を最大のもので20個格納する。このリストはAndroid上のSQLiteデータベースに格納され、全ての要素に対してアクセスすることができる。また、一つの場所に対してMACアドレスが複数取得できるが、多くは電波が弱いいため今回取得上限を20とした。

表6 場所モデルのデータベース構造

部屋のID	部屋名	取得回数	MAC 1	RSSI 1	MAC 2	RSSI 2	...	MAC20	RSSI20
1									
2									
⋮									

4.3 場所モデルの学習

各場所で端末のみでMACアドレスとそのRSSIの強度、取得回数をカウントし、電波強度の和を取得回数でそれぞれ除することでその部屋の重心を求める。学習サンプルはユーザーがその場所に訪れることで学習することができ、過去のサンプルに関係なくデータの追加を行うことができるため、再学習において端末の負荷は少ない。

4.4 提案方式での場所検知方法

3.2.3で提案した方式の場所検知の実現方法について構成する三つのステップと各ステップで用いる計算式について具体的に説明する。

場所検知アルゴリズムは、次の3つのステップで構成されている。

- STEP1.場所候補リストの抽出
- STEP2.各場所候補に対して重心を計算
- STEP3.重心との距離の差を利用して場所検知

STEP1. 場所候補リストの抽出

場所候補のリストは、端末がスキャンして検出したAPが観測される全ての場所を含んでいる。この場所候補リストは、APのMACアドレスをキーとして、表6のデータ構造を持つデータベースを検索して得られる。ここで、WiFiのスキャンデータを

$$s = [r(a_0), r(a_1), r(a_2) \cdots r(a_k)] \cdots \textcircled{7}$$

とする。 a_k はアクセスポイントのMACアドレス、 $r(a_k)$ はアクセスポイント a_k のRSSIである。ここではMACアドレ

スに重複はあるが、取得できたものをそのまま表示している。これらを端末内で読み込み、各 MAC アドレスの RSSI の強度の平均を計算する。アクセスポイント a_k が観測される場所のリストを次式で表す。

$$L(a_k) = \{l_1, l_2, \dots, l_n\} \dots \textcircled{8}$$

ここで l_n はアクセスポイント a_k が観測される第 n 番目の場所を表す。⑦と⑧の結果より WiFi のスキャンデータ s を観測した時の場所候補は、

$$C(s) = \{c_0, c_1, \dots, c_m, \dots, c_M\} \dots \textcircled{9}$$

と定義する。 c_M は $C(s)$ の第 m 番目の場所候補である。

STEP2. 各場所候補に対して重心を計算

各場所候補 c_M について重心を計算する。はじめにスキャンデータ s に対し、重複した MAC アドレスの RSSI の平均化を行う。ここで、場所 l で観測される AP の MAC アドレスおよび RSSI の平均のリストを

$$A(l) = \{a_1, r_1, a_2, r_2, \dots, a_n, r_n, \dots, a_{N(l)}, r_{N(l)}\} \dots \textcircled{10}$$

とする。 a_n は第 n 番目の MAC アドレス、 r_n は a_n の MAC アドレスの平均値であり、 $N(l)$ はリストのメンバ数を示す。

STEP3. 重心との距離の差を利用して場所検知

場所候補 $C(s)$ に対し、 $A(l)$ の値を用いてスキャンデータと学習データの重心の距離 X_m を計算する。場所候補 $C(s)$ の要素はそれぞれ $A(l)$ と同じデータ構造を持っているため、同一の MAC アドレスに対し、二乗誤差を計算する。

それによって得られた結果を場所候補ごとに合計し、得られた合計値を X_m とすると

$$\operatorname{argmin}\{|X_0|, |X_1|, \dots, |X_m|, \dots, |X_M|\} \dots \textcircled{11}$$

となり、これによって得た m を取得し、スキャンデータによる部屋の識別結果を c_m であると判定する。

4.5 評価実験

4.5.1 評価項目

本研究では、提案手法による場所検知の正解率を評価項目とする。ここでは正解率を、場所 A でスキャンした全 WiFi データのうち場所 A として検知した回数の割合と定義する。

4.5.2 データ収集

データ収集者がデバイスを持ち、千葉工業大学津田沼キャンパス 6 号館の全部屋に入り、部屋の中で WiFi データを採取した。各場所で 5 秒間 150~250 個程度のデータを取得し、データベース上に格納する。ここで、「個人での利用を目的としている」という運用方法を想定して、実験のデータ収集は全部屋のデータ取得したのちにテストデータを 3 部屋分取得した。選んだ部屋は現在講義で自身が使用している 615, 632, 641 教室である。

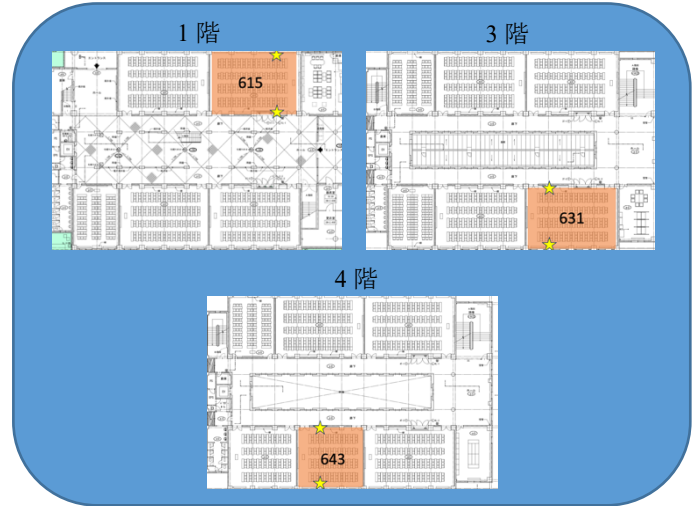


図 10 実験を行った対象の部屋
(千葉工業大学津田沼キャンパス 6 号館)

4.5.3 学習と場所検知テスト

学習データを格納したデータベースと、取得しておいた 3 部屋分のテストデータを用いて場所検知テストを行う。実験の場所検知正解率を用いて正解率の計算も行う。

4.5.4 結果

表 7 は実験結果を示しており、入力が今回選んだ部屋、判別がその入力データに対する各部屋の番号を示しており、内部の数字は入力データの重心との誤差の合計をそれぞれ示している。そして誤差合計が一番少ないものが今回の判別結果としている。数値が入力されていない部分に関しては重複する AP が観測されなかったため、場所候補から外している。図から分かるように 615, 631, 643 はそれぞれ正しく場所検知を行うことができた。また、識別に要する時間はデータスキャンの 5 秒とは別に、1 秒未満で部屋候補との重心比較計算を行うことができています。

表 7 場所検知テスト結果

入力	判別	611	612	613	614	615	621	622	623	624	625
615		44013187	68074779	76052900	24061300	35870	833809	*	*	*	44331321
631		56218382	*	*	*	*	258563	*	*	*	*
643		*	*	*	*	*	*	48306497	68110004	*	*
入力	判別	625	626	631	632	633	634	635	636	642	643
615		44331321	825804	36649521	68015806	*	*	*	44143712	*	*
631		*	*	5902	64145506	*	*	*	*	32440813	68236005
643		*	*	76052441	12599809	36477184	*	*	*	44396019	25806
入力	判別	644	645	646	647	652	653	654	655	656	
615		*	*	*	76073984	*	*	*	*	*	
631		*	*	*	*	60299995	64095021	*	*	*	
643		44508565	*	*	*	64234891	8772112	40427029	56379900	*	

5. 考察

第 4 章で行った実験の結果により、重心比較法を用いることによってサーバーを使わずに部屋識別をすることが可能であることがわかった。また、実験を通して取得場所によって取得できる MAC アドレスの数に違いがあることで、取得環境の MAC アドレス当たりのデータ取得数に違いが

あることが分かった．そこで MAC アドレスの数が異なる部屋データを取得したところ、結果は表 8 のようになった．

表 8 部屋別の MAC アドレス当たりの取得回数

	部屋 1	部屋2
観測された AP の数	13	41
AP 当たりの取得回数	20	6

表 8 は取得時間が同じ場合で、場所のみを変えてデータの取得を行った時の MAC アドレス当たりの取得回数を表したものである．表から分かるように、その部屋で観測することができる MAC アドレスの数が約 3 倍の取得量となると、MAC アドレス一つ当たりの取得回数が約 3 分の 1 となっている．このことから、一定時間内に取得ができる MAC アドレスの数は大体決まっており、データの総取得量は変わらないということがわかる．そこでデータスキャンは時間単位で決めるのではなく、観測することができた MAC アドレスの数に対して動的に変化させる必要があると考えた．

次に、今回データを取得する際は人がいない状況でのデータ取得をしたが、人がいる場合ではデータの数値に影響があるのではないかと考え、講義室の席数に対し約 60%の在席者がいる場合とそうでない場合とを比較した．また、同時に部屋のデータ取得場所を変え、差があるかどうかと比較した．

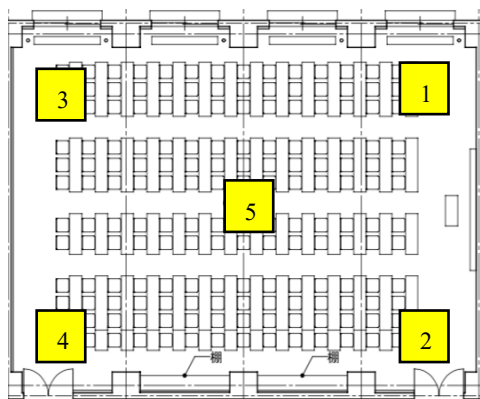


図 11 部屋の取得位置

表 9 人がいる場合の AP の RSSI 平均値

位置	1	2	3	4	5	1-5	学習データ
RSSI 平均	61.78	65.18	64.44	55.33	57.56	60.86	58.60

図 11 は表 9 と連動しており、図 11 の数字で示している部分が表 9 の「位置」に対応している、その場所でデータを取得している．また、この部屋での影響度が大きい単一の MAC アドレスに対してその場所で取れた RSSI の平均値を示している．表 9 の学習データは人がいない場合の同一の

部屋のデータである．このことから部屋の中で場所が違い、人がいたとしても AP から取得ができる RSSI の平均値はあまり変わらないため、データ取得に関して影響は少ないと考えられる．

6. まとめ

本研究では、「ランドマーク単位で自分の位置を認識」でき、「極力軽量の計算量」でかつ「学習データが増加しても処理量が変化しない」特徴を持った位置推定技術の提案を行った．個人情報に他を漏らさない方法として WiFi という既存のインフラを用いることで、サーバーにユーザー情報を送らずに処理をする部屋識別システムを提案した．ランドマーク単位で位置を識別するには場所固有の AP を取得することができれば利用可能である．また「重心比較法」を用いることにより、位置推定に要する計算量の軽量化をすることができ、再学習をする際に処理量が変化しない．

端末を用いての評価実験では取得したデータの場所と推定結果の場所は一致しており、提案手法の再現ができていた．全体の処理時間に関しても十分に短いと言えるものであるため実時間での実装は可能であり、5 章で行った実験の結果から部屋の中での取得場所に関しても問題はないと考えられることが分かった．

今後の展望としては、今回の提案では部屋の推定結果を表示するのみであり、サービスとしての利便性が欠如している．そこでランドマーク単位での識別を利用して、取得した AP によって端末上のアプリケーションの並びを変更し、その場所で推奨するアプリを表示することができるような、サービスを考えていきたい．

参考文献

- [1] Matt Duckham and Lars Kulik. Location privacy and location-aware computing. Dynamic & Mobile GIS: Investigating Changes in Space and Time, pp. 34-51, 2006.
- [2] 中澤啓介, 北望, 高木健士, 井上智雄, 重野寛, 岡田謙一, ランドマークの視認性に基づいた動的な案内地図作成, 情報処理学会論文誌, Vol.49, No.1, pp.233-241(2008)
- [3] WWDC2013
<https://developer.apple.com/videos/play/wwdc2013/307/>
- [4] Anthony LaMarca, Jeff Hightower, Ian Smith1, and Sunny Consolvo. Self-mapping in 802.11 location systems. In UbiComp 2005: Ubiquitous Computing, Lecture Notes in Computer Science, pp. 87-104. Springer Berlin / Heidelberg, 2005.
- [5] 暦本純一, 塩野崎敦, 末吉隆彦, 味八木崇, PlaceEngine: 実世界集合知に基づく WiFi 位置情報基盤, インターネットコンファレンス 2006, pp.95-104 (2006).
- [6] AirMac ユーティリティ
<https://support.apple.com/ja-jp/guide/aputility/welcome/mac>