

加速度センサーを用いた BLEの3点測位の精度改善

秋山 佑太¹ 高木 優希² 串田 高幸¹

概要: 近年、位置情報を使用する IoT デバイスが増加している。GPS が屋外で使われている中、屋内では GPS が届かない場合がある。そのため、GPS の電波が届かない場所におけるデバイスの位置情報の特定が必要である。本研究では低消費電力で動作する ESP32 を使い BLE と加速度センサーで測位を行う。BLE の 3 点測位だけでは誤差があり、正確な位置を取得するのは困難である。また、マップに表示した場合に壁の中や障害物の上に表示されてしまうことがあり、正確な位置が把握できない。そこで加速度センサーで歩数を測定し、歩数を距離に変換する。本提案では、BLE の RSSI で取得した距離で 3 点測位を行い、取得した 3 点測位の座標を加速度センサーで修正し、精度改善をする。評価として既存の BLE だけで行う 3 点測位の誤差と、本提案の誤差を比較する。

1. はじめに

背景

近年、IoT(Internet of Things) デバイスの普及が進み、飲食店や公共施設、一般的な家庭に IoT が使われ、人々の生活を支えている [1]。IoT デバイスには WiFi や Bluetooth が搭載されており、それらの電波を利用してデバイス間やサーバにデータを送信する役割がある。その中で GPS や Bluetooth の LE(Low Energy) が使われることにより、身近なものでわかりやすいものとして GPS(Global Positioning System) が有名である。

GPS は GPS 衛星を使った位置情報を特定するものである。まず、GPS を搭載し使用している端末の GPS 受信機が電波を受信する。次に端末側の受信機は送信機から送られてきた信号が届くまでの時間に電波の速度を乗算し、送信側の衛星と受信側の端末の距離を推定する。そして送信側の衛星の 4 台から取得した距離をもとに自分が使用している端末の現在位置を特定するものである。GPS は位置情報を手軽に取得できることで多くの車や端末で使われている。しかし地下通路やデパートの地下、屋内の場所では GPS の信号が微弱になるため、図 1 のように現在地が取得できないことや測位に時間がかかる場合がある。なお、図 1 は、複数階層の建造物内での位置の誤差について示して

おり、GPS は位置のずれが発生するが、BLE では屋内に配置されたビーコンを用いるため、高精度な位置測位が可能である。

このように、問題の対策として地下通路やデパートの地下、屋内の場所で正確に位置情報を取得できる技術が必要となる。屋内測位の例として、BLE を使ったビーコン測位や RFID タグを使った近接無線通信が挙げられる [2]。また、位置情報を取得する方法やスマートフォンを使った加速度センサーで歩行者の歩数を推定して歩幅を計算し、移動距離を算出できる [3]。その中でも BLE ビーコンと加速度センサーを使った方法は導入コストが低く、消費電力も低いいため、屋内測位で使用されることが多い [4]。

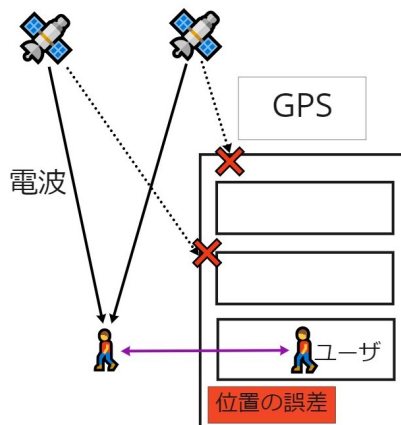


図 1 位置の誤差

¹ 東京工科大学コンピュータサイエンス学部
〒192-0982 東京都八王子市片倉町 1404-1
² 東京工科大学大学院 バイオ・情報メディア研究科 コンピュータサイエンス専攻
〒192-0982 東京都八王子市片倉町 1404-1

速度の変化を測定する加速度センサーは加速度を計測し

て移動距離を推定するものである。加速度センサーの問題点として、移動距離は推定できるが移動方向は推定できない。そのため、別の測位する方法と組み合わせて精度を改善する必要がある。BLE の測位方法としてビーコンを使った近接測位と 3 点測位が挙げられる [5]。近接測位では 1 つのビーコンを用いて距離が近いビーコンを判別し、データをサーバに送信および分析をすることで現在位置を取得する方法である。この方法では詳細な位置は取得できず、誤差 100cm の範囲しか取得できない。

一方 3 点測位では、まず、3 つ以上のビーコンを使用し、電波強度である RSSI をそれぞれ取得する。次に RSSI の値を距離に変換し、この距離を半径とする円を作成する。これらによって作成された 3 つの円の交点を計算することで座標として位置が求められる。3 点測位では、絶対値によって取得できるため、近接測位の電波強度のみで測位する方法よりも条件によって精度が高くなる。しかし、障害物や人によって RSSI の値に誤差が発生する。そのため BLE だけでは正確な取得が難しい場合がある。そこで別の位置情報を取得する方法と測位を合わせて精度を改善する方法が必要である。

課題

背景で述べたように、GPS では地下通路やデパートの地下、屋内といった電波が届かない場所に設置されている場合、正確に位置情報を取得することが難しい。そのため、屋内で位置情報を取得可能にする場合は、屋内測位技術を採用する必要がある。

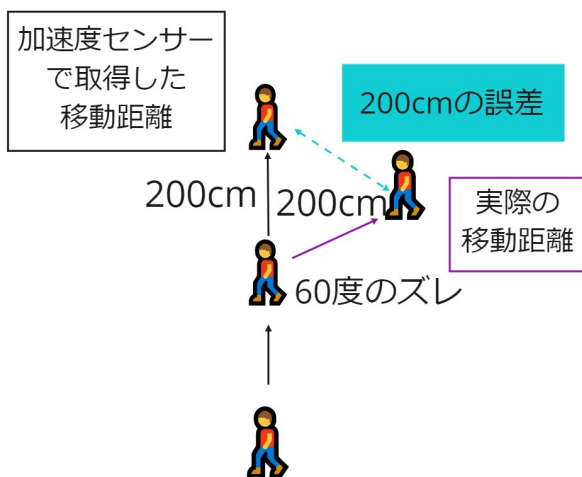


図 2 加速度センサーの誤差

まず、ラーニングフェーズとしてユーザから身長を入力してもらい、加速度センサーで歩数を推定することによって歩幅を計算する。歩幅を求めることによって、移動距離が計算可能である。しかし、歩数や歩幅に誤差発生すると、移動距離にも誤差が生じる。また、図 2 のように横に移動する場合、加速度センサーは距離のみ計算可能であるため、

60 度ずれて 200cm 進んだ場合、200cm の誤差が起きる [6]。

また、BLE を用いた測位手法のうち、絶対値で距離が求められる点から 3 点測位に焦点を当てる。3 点測位の問題点は、誤差のばらつきが大きいことである [7]。基礎実験として、ビーコンの位置を 300cm 間隔で設置し、位置情報の測定を行った。RSSI の値から距離を求めるために、以下の式を用いた [8]。なお、伝搬指数である n の値は 6.5 を採用した [8]。

$$\text{距離} = 10^{(TxPower - RSSI)/(10 * n)}$$

TxPower(dBm) は送信側の持つ電波強度である。RSSI(dBm) は受信した無線信号に存在する電力の測定値である。そのため、基礎実験で取得した 100cm での RSSI の最頻値を入力する。実際の位置はメジャーで測定した。実際の位置はデバイスの位置で、計算で求められた距離は 3 つのビーコンを使った 3 点測位の位置である。実験で座標の位置の誤差を測定した結果、誤差が 56cm の場合の実際の位置と計算によって得られた位置を表 1 に示す。

表 1 誤差が一番少なかった場合の結果 (cm)

実際の位置	計算で求めた位置	ユークリッド距離
(170, 200)	(152, 148)	56

実際の位置と計算によって得られた位置を比較すると、従来の 3 点測位だと 56cm より小さい誤差で測定することができない。誤差が大きいことで、マップに表示した場合に壁の中や障害物の上に表示されてしまうことがあり、正確な位置が把握できない。したがって、展示会やリアル脱出ゲームのようなルートを表示するような場合では、正規のルートで動いているかどうかを把握することができない。

各章の概要

第 2 章では関連研究について述べる。第 3 章では提案するシステムについて述べる。第 4 章では提案したシステムの実装と実験方法について述べる。第 5 章では実験の評価手法と分析手法を述べる。第 6 章では提案したシステムの議論を述べる。第 7 章では本研究のまとめを行う。

2. 関連研究

関連研究では、課題を既存方法を 4 つ取り上げ、どのように解決しているかについて説明を行う。

1 つ目に画像認識技術を使った方法がある。カメラによってキャプチャし、画像から距離を推定し、座標を導き出すというものである [9]。低コストで画像処理を行い、移動ロボットでラベル ID をつけて座標を検出し、低コストで測位を行うことができる。

2 つ目に超音波を用いた位置特定システムを挙げる [10]。従来の超音波では電波干渉によって、測定が難しい場合が

ある。しかし、超音波信号の角度制限と電波の干渉を最小限に抑えてロボットを使うことによって精度が向上している。しかし、ポーリングによって測定値を収集しているため、ポーリングプロセスが起ってしまう。結果、データの収集率が悪くなり、ロボットの座標を絶えず結合する必要がある。

3つ目に加速度センサー・ジャイロセンサー・地磁気センサーを用いた歩行者推測航法 (PDR) がある。PDR だけでは誤差が起きてしまうがアルゴリズムで誤差を減らす方法がある [11]。GPS が劣化した場合に、歩数の長さや方向が PDR で重要な問題である。拡張カルマンフィルタと2軸のコンパスを使って提案された結合モデルで方位を補正している。しかし、磁気コンパスで予想可能な誤差を減らしているため、磁気擾乱が多い環境では正確な方位を取得できない。

4つ目に BLE を用いた屋内測位システムがある。BLE の誤差を閾値 k と類似性メトリック関数を使い、精度改善を行っている。結果、推定位置の誤差は約 1.8m である [12]。課題として動的に受信して補正ができない、BLE 以外のデバイスを組み合わせて測位できない。

3. 提案方式

基礎実験

提案方法を成立させるため、基礎実験として BLE を用いた3点測位と加速度センサーを用いて歩数を計測し、それぞれで計算した位置と実際の位置の比較を行なった。データを取得するセンサーの動かし方について図3に示す。

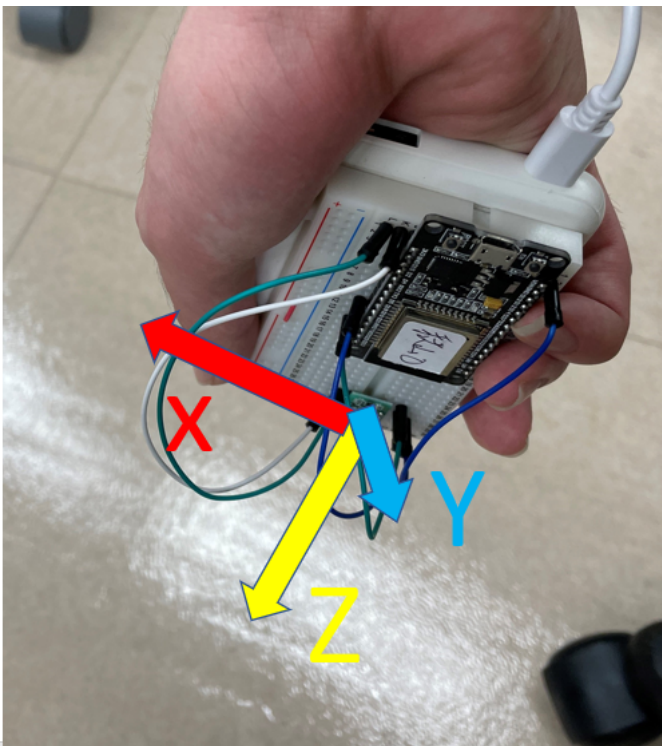


図3 センサーの持ち方

加速度センサーは図3のように、Z軸を下に向けてX軸を前後に振るようにして行う。本研究では500cmの直線でデータを取得しているため、173cmの男性が歩く1歩当たりの歩幅は以下の式によって求められる。

$$\text{歩幅} = \frac{\text{移動距離}}{\text{歩数}}$$

歩幅を求めるために歩数を決める必要がある。そのため、500cmを歩いて歩数を計測する必要がある。図4のように500cmの直線を歩いて歩数を計測する。

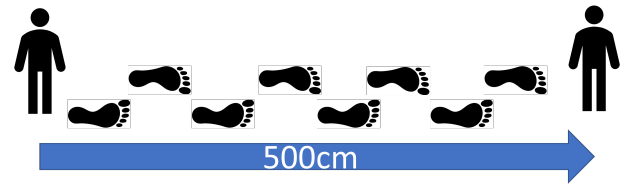


図4 歩数の計測

173cmの男性の場合、500cmで計測した結果、8歩で歩幅は62.5cmであった。しかし、加速度で測定するにあたって、毎回、500cmを歩いてもらい、歩数を測定することは困難である。ここで、移動距離が不明な場合でも歩幅を測定する方法として歩行率がある。歩行率は以下の式によって計算される [9]。

$$\text{歩行率} = \frac{\text{歩幅}}{\text{身長}}$$

そのため、移動距離が分からない場合でも、歩行率を得ることができれば身長のみで1歩の歩幅が推測できる。しかし、人によって歩幅が変化するため、歩行率を定義する必要がある。そこで平均170cmの男性8人に5mの直線を歩いてもらい、同じぐらいの身長でどの程度誤差があるのかを測定した。500cmでの歩数の差は最大で2歩であることが分かった。

歩数が同じ161cmと170cmでも、161cmは0.44step/m、170cmは0.42step/mで歩行率は0.02step/mの誤差がある。これは歩幅を身長で除算して求めるため、歩数が同じ場合、身長に関係なく歩行率の差が発生する。また、175cmは歩数が6歩だったため、歩行率が0.48step/mで最大値になっていることが分かった。本提案では、これらの平均値を求め、0.4steps/mを歩行率の値として採用した。

図5では身長によって歩幅がどのように変化しているかを示す。歩幅の関係性では歩行率と同じ形式のグラフが表示された。図5から歩幅は個人の歩き方に依存するため、歩幅と身長に相関関係がないことがわかる。そのため、ランニングフェーズで1人1人歩幅の測定する必要がある。

次に1歩をどのようにして判定するかを定義する。実験により取得できた数値を図6に示す。図6では173cmの男性が腕にセンサーを取り付けた状態で500cmの直線を歩

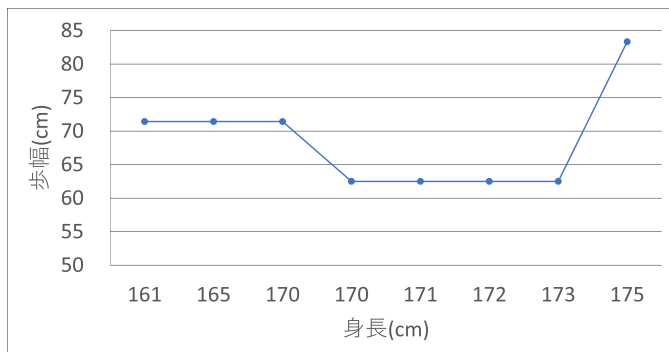


図 5 身長と歩幅の関係性

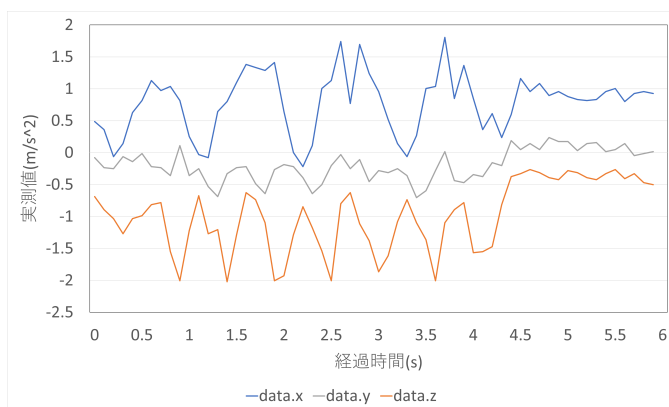


図 6 加速度センサーから取得した値

行時の加速度センサーの変化量である。

ここで、どのようにして加速度センサーで取得したデータから1歩と判定するかについて説明を行う。加速度センサーで取得したデータから1歩と判定するには閾値を決める必要があり、図6のグラフからdata.xとdata.yは歩数として相関がなかったため、data.zのデータを使用する。本研究では加速度センサーで歩数を取得するため、変化が大きいことから、data.zにおける下に凸の回数を歩数として定義する。また、data.xとdata.yは歩数として計測できなかったため、本研究では使用しない。data.zは腕にセンサーを装着した場合、腕を振るタイミングで $-2m/s^2$ の数値を取得することが分かった。しかし、data.zの0.5sの取得時点では、測定地点から $0m/s^2$ の状態で動き出しているため、途中の腕を振っている場合と比べて値の振りが少ない。次に図6と腕を振っているタイミングでどこが1歩なのかを測定する。data.zの0.5sの取得時点では、腕は図7の状態になっている。図7では腕を後ろに振り上げる形になっているため、0.1sから0.7sまでの値の中では最小値である。

次にdata.zがの1sの取得時点だった場合、腕は図8の状態になっている。図8では腕を振り下げる形になっているため、0.7sから1.3sまでの値の中では最小値である。

次にdata.zの1.2sの取得時点だった場合、腕は図9の状態になっている。図9では腕を振り上げて、最高地点になっ

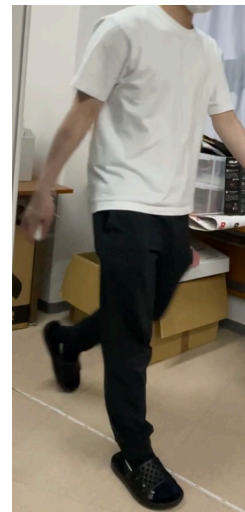


図 7 data.z の 0.5s の取得時点での腕の状態

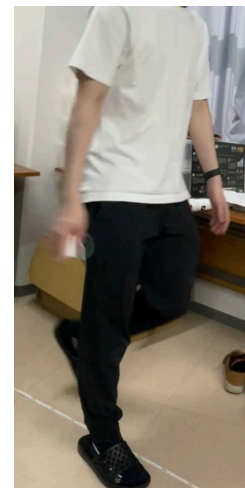


図 8 data.z の 1s の取得時点での腕の状態



図 9 data.z の 1.2s の取得時点での腕の状態

ているため、0.9sから1.7sまでの値の中では最大値である。以上のことからdata.zの変化で1歩だと推定することは

可能であることが分かった。

次に1歩当たりの閾値をどのようにして定義して歩幅の誤差がどのくらい出てしまうのかを図10に示す。aからhは各身長であり、aは161cm、bは165cm、cは170cm、dは170cm、eは171cm、fは172cm、gは173cm、hは175cmを示している。この閾値はdata.zの変化値で1歩だと判定するための値である。歩数の誤差が小さければ、誤差が小さいことになる。

閾値が $-1.14m/s^2$ だった場合に、歩数の誤差が小さくなる。そのため、閾値を $-1.14m/s^2$ とする。

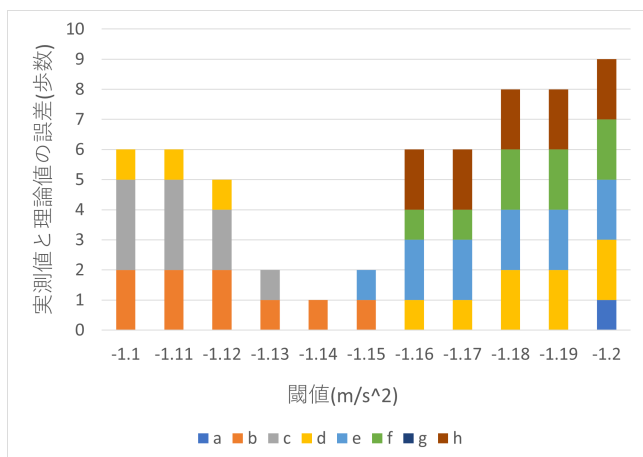


図10 閾値と誤差の関係

また、8人の合計歩数が59歩で誤差が1歩であるため、認識率は98%あることが分かった。しかし、8人で得られた閾値であるため、ユースケースで使用するには、ランニングフェーズで500cm歩いてもらい、歩幅と閾値を決める必要がある。そのため、閾値は500cm歩いてもらい、歩幅と $-1m/s^2$ から $-1.2m/s^2$ の間で認識率が高い値にする。

次にBLEの基礎実験では、100cmの基準値を決める測定を行った。図11では2つのビーコンを使用し、ビーコンの間隔を100cmにし、測定を行った結果である。最頻値である $-65dBm$ が100cmの基準になる。しかし、次に多い取得回数として14回目の $-73dBm$ がある。そのため、RSSIの誤差範囲は $-10dBm$ で、70cmの誤差がある。また、設置場所が異なると図12のように値が変化する。以上のことから、測定する場所によって値が変化するため、ランニングフェーズでビーコン1つ1つ基準値を決める必要がある。

提案方式

BLEと加速度センサーの組み合わせた測位方法を提案する。提案の条件として、BLEのビーコンは固定して行う。次にセンサー搭載デバイスは人によって移動されるものとし、人は等速直線運動(1m/s)で動くものとする。また、ランニングフェーズでユーザーがBLEの100cmのRSSIの基準値を決める。基準値を決める際の取得回数は200回と

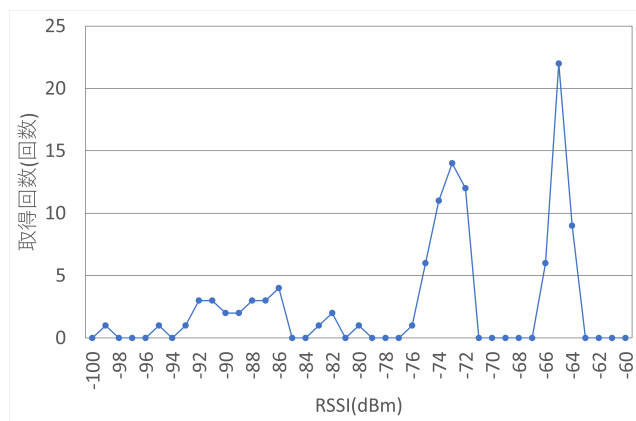


図11 100cmのRSSIの値

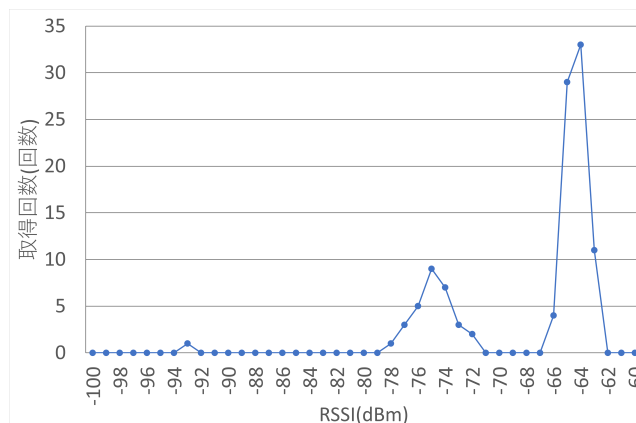


図12 別の場所で取得した100cmのRSSIの値

決める。また、ビーコン1つ1つの精度を決めるため、取得した値が密集している山を探索する。図11を元に基準値を算出する方法を図13を用いて説明する。

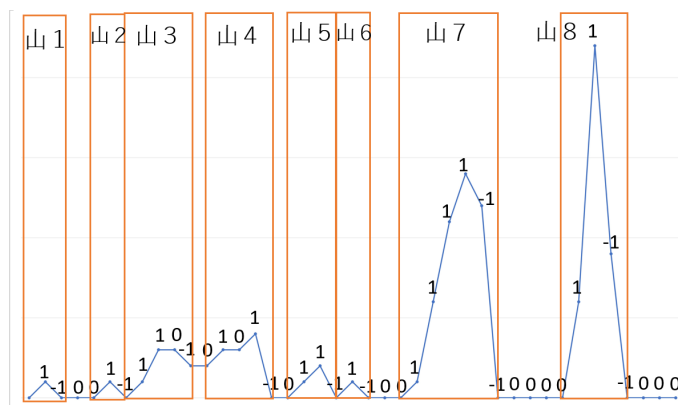


図13 基準値の計算方法

まず、図11の分布図から山がいくつあるのかを把握する。RSSIの値が増加した場合に1、減少した場合に-1、変化がない場合は0として数値を置き換える。したがって、-1と0あるいは-1と1という組み合わせができた場合に、その地点を山の区切りとして判断する。図13の場合、山が8つあることになる。この山の数を基準にビーコン1つ1つ

の精度を決める。RSSI の基準値は最頻値で決める。加速度センサーによって得られた加速度から距離を計算する過程で用いる身長は、ユーザによって入力されるものとする。

基礎実験により、1つのビーコンの測位する RSSI より加速度センサーで取得した値の方が精度が良いことが分かった。それにより、図 14 ように加速度センサーで取得した 1 歩が測定されたタイミングで、加速度センサーで推定された値に近い RSSI を参照する。

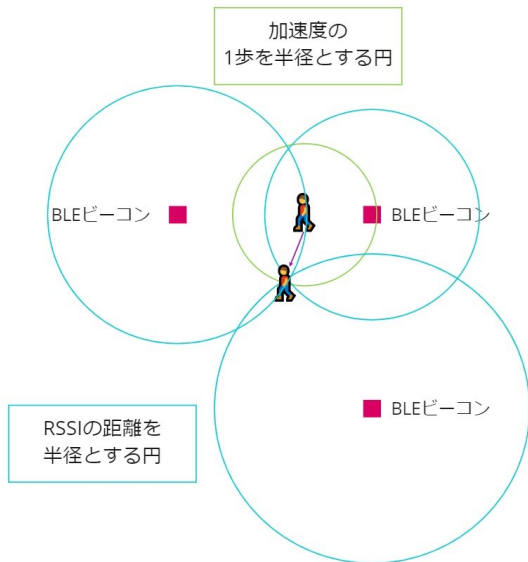


図 14 提案方式

参照した 3 つのビーコンを用いて 3 点測位を行う。3 つで取得した RSSI で 3 点測位を行い、加速度センサーで座標を修正する。一歩ごとに 3 点測位を繰り返し行い、2 歩目以降からは 1 つ前に算出された座標を基準にして次の位置の座標を求める。図 15 では 3 点測位の方法を示す。

x, y を求める式を以下の式に示す。

$$y = \frac{v_b(x_c - x_b) - v_a(x_a - x_b)}{(y_a - y_b)(x_c - x_b) - (y_c - y_b)(x_a - x_b)}$$

$$x = \frac{v_a - y(y_c - y_b)}{(x_c - x_b)}$$

上の式を用いて x, y を求めることができ、座標として表示できる [7]。また、本研究では 1 つの円を加速度センサーの 1 歩で半径と計算する。3 つのビーコンを使用し、3 点測位を行う。その後、加速度センサーの 1 歩の半径と最も近い値に座標を補正する。図 16 は図 15 に加速度を追加した補正方法である。

図 15 に図 16 のように加速度で決めた円を追加して、3 点測位で求めた座標を加速度で求めた半径に最も近い値の座標に補正する。スタート地点から加速度で 1 歩を検出した際に、3 点測位で座標を算出する。その後、加速度を 1 歩とする円で補正を行う。3 点測位を行う 3 つのビーコンの基準として、図 13 で説明した山を基準に決める。図 12 で

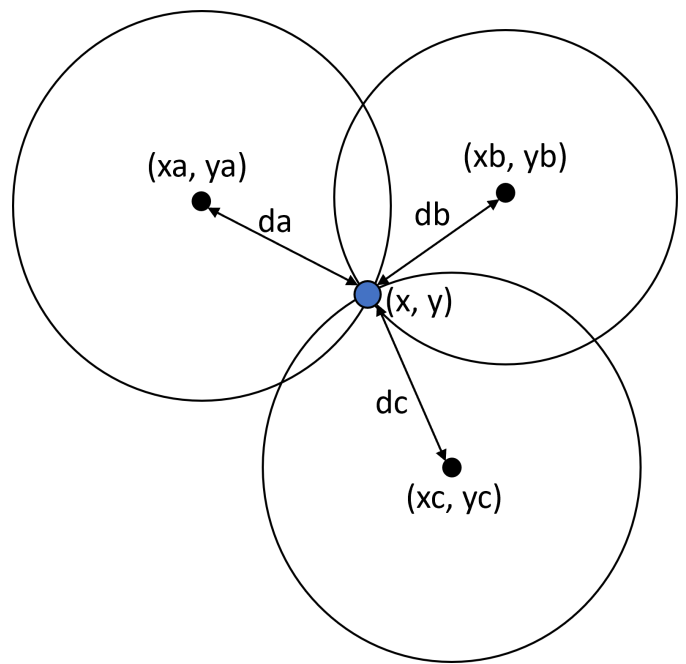


図 15 3 点測位の方法

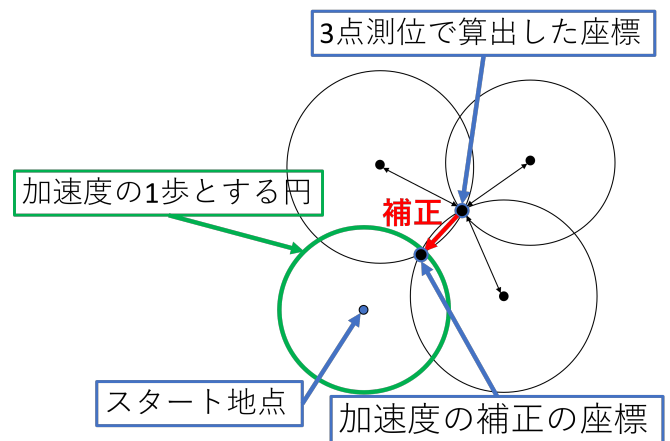


図 16 3 点測位の補正方法

は山の数に 3 つに対して、図 11 は山が 8 つである。ビーコン 1 つの精度として図 12 の方が精度が良い。したがって、3 点測位する際に付近に 4 つビーコンがあった場合に 2 つのビーコンが決まってい、もう 1 つを決める基準として、図 11 ではなく、図 12 を 3 つ目のビーコンの 1 つとして決める。

システム全体の構成図を図 17 に示す。図 17 の IoT デバイスでは搭載されているセンサーから取得された加速度および RSSI を Web サーバに送信している。web サーバでは取得されたデータをマップへ反映している。ユーザはブラウザを通して視覚的に位置がわかる。この流れを詳しく説明する。

まず、IoT デバイスから取得した RSSI と加速度センサーから取得した加速度を web サーバのデータベースに送信する。次に web サーバではデータベースに格納した実測値

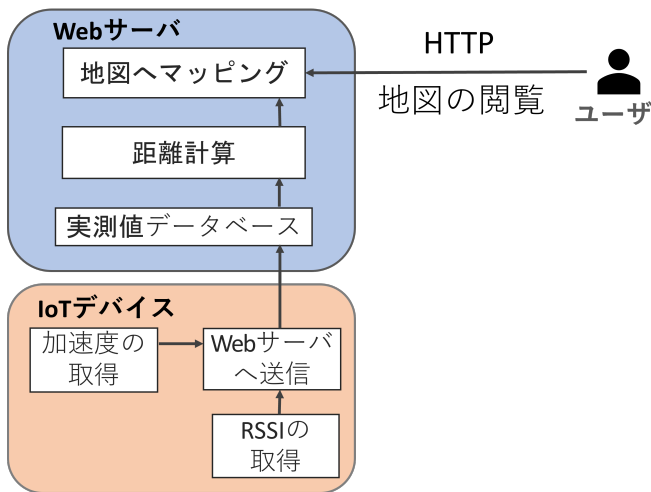


図 17 システムアーキテクチャ図

を用いて距離への変換および位置の座標を計算する。その後、マップへ反映する。ユーザは HTTP 通信を用いてマップの閲覧を行い、位置を確認する。

ユースケース・シナリオ

本提案のユースケースは、BLE ビーコンが配置済みである地下通路やデパートの地下、展覧会やリアル脱出ゲームといった、リアルタイムで正確な位置情報を知りたい場合である。BLE の RSSI 値を用いた 3 点測位のみでは、遅延時間や電波の反射・吸収・干渉によって図 18 のように誤差が発生し、正確な位置をリアルタイムで取得できない。

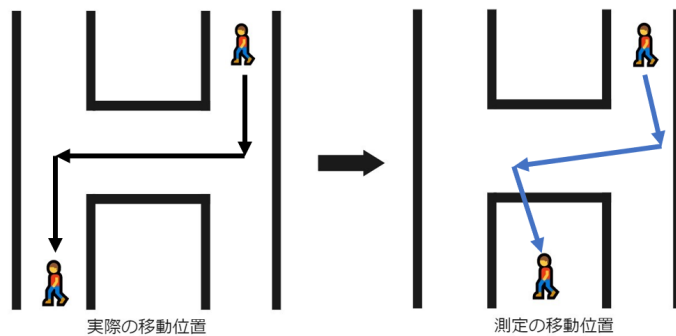


図 18 測定位置の誤差

電波は、周囲の障害物や閑居によって影響を受けやすく、BLE の 100cm の基準値も変動する。対策として、100cm の時の RSSI 値を把握するラーニングフェーズを行う。100cm の時の RSSI 値を複数箇所によって取得し、1 つ 1 つのビーコンの基準値を決定することで、本提案を導入する周囲の環境に適合させることが可能である。加速度センサーは、距離を算出するために必要な要素は身長と歩行率である。身長はユーザの手によって入力されるものし、歩行率はユーザに歩いてもらうことで算出する。ユーザ自身に 500cm を歩いてもらい、歩幅を求め、歩行率を計算する。これらのラーニングフェーズをもとに BLE および加速度

センサーを用いた位置の座標計算が可能となる。ユーザは単に歩くのみでマップに自動反映されるようになる。

4. 実装と実験方法

実装

本研究で使用した機器は BLE ビーコン、IoT デバイス、加速度センサーである。BLE ビーコンおよび IoT デバイスとして ESP32 を 4 台使用し、加速度センサーとして BMX055 を使用した。デバイスの位置情報の解析する web サーバとして OS は Ubuntu20.04 の仮想マシンを用いた。サーバソフトウェアとして Nginx、使用言語は micropython である。実装するにあたって、ラーニングフェーズで加速度の閾値と RSSI の基準値を決める必要がある。まず図 20 にシステム構成図を示す。

図 20 は図 17 で示したシステムアーキテクチャをもとに実装するシステムである。IoT デバイスの位置情報を解析するために、加速度センサーのデータと RSSI データを格納するデータベースとして Mongo DB を採用した。受信側の ESP32 が他の ESP32 からの電波を受信し、RSSI 値を取得する。加速度センサーは 1 歩を検出した際に RSSI の値を Mongo DB へ送信する。この時、各 BLE ビーコンに番号を付与し、Mongo DB の database1, database2, database3 と実測値の格納先を分けて送信する。受信側の ESP32 には BMX055 を搭載して加速度を取得し、Mongo DB の database4 に送信する。加速度センサーは常時加速度を取得している。また、BLE ビーコンも同様に電力供給が続く限り RSSI の値を取得し続けるものとする。learn database はラーニングフェーズで取得したデータを格納するデータベースである。図 19 はラーニングフェーズで取得したデータの加速度の閾値と RSSI の基準値を決めるシステムである。2 つのデータを計算し、learn database に格納する。

サーバ側ではマップ上における座標の計算は加速度センサーの 1 歩が測定され、データベースに RSSI の値が格納されたタイミングで実行され、3 つのビーコンで取得した 3 点測位に加速度センサーで補正をする。マッピングは web ブラウザに座標を表示させる。使用するプログラムは Flask を用いて web ブラウザに表示させる。

座標を反映するマップは事前に作成するものとし、ユーザはブラウザを通して位置が確認できる。

サーバの実装

サーバのソフトウェアについて説明する。

図 21 ではサーバで実装するソフトウェアの構成を示している。Mongo DB の database は必要なデータ数だけ設置されている。Web サーバでは Nginx を使用する。Web アプリケーションは Python のフレームワークである Flask を使用して、Web ブラウザ上のマップにプロットしたデ

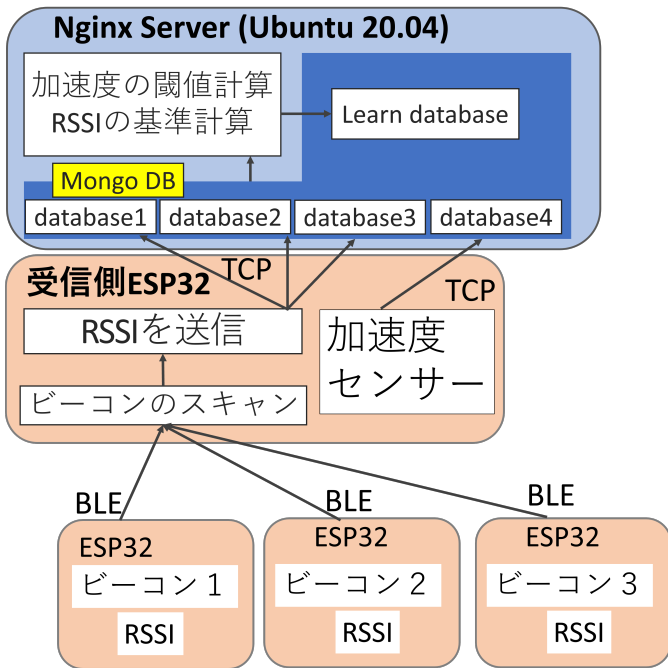


図 19 加速度の閾値と RSSI の基準値のシステムの実装

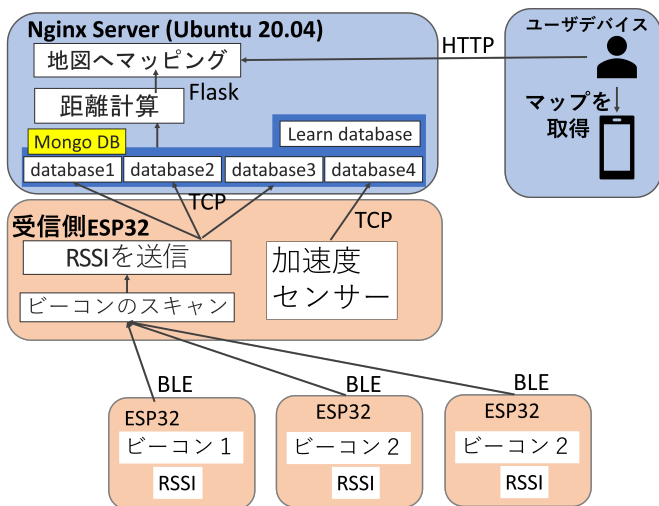


図 20 システムの実装

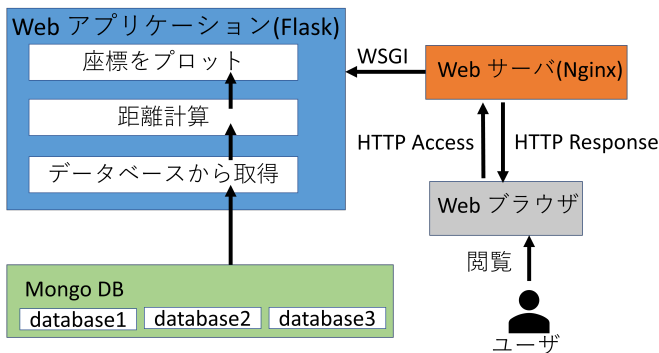


図 21 サーバのソフトウェア実装

を表示する。Web サーバと Web アプリケーションは WSGI で通信をする。座標のプロットのプロセスは

HTML5 と Flask で実行する。

実験環境

本研究の実験では東京工科大学の研究棟 A910 で行い、GPS が届かない場所と定義し、実験をする。RSSI を取得するために受信するデバイスとして ESP32 を使用し、送信するデバイスとしても ESP32 を使用した。RSSI を取得する際にできるだけ障害物をなくすために高さ 2 m に ESP32 を設置した。加速度を取得するためにセンサー BMX055 を使用した。ソフトウェア環境では送信側のデバイスの ESP32 に BMX055 の座標と RSSI を取得する micropython でコードを組み込み、受信側のデバイスの ESP32 に RSSI を取得するためにも同様に micropython をコードで組み込んだ。取得した BMX055 の加速度と RSSI を Mongo DB に保存する。

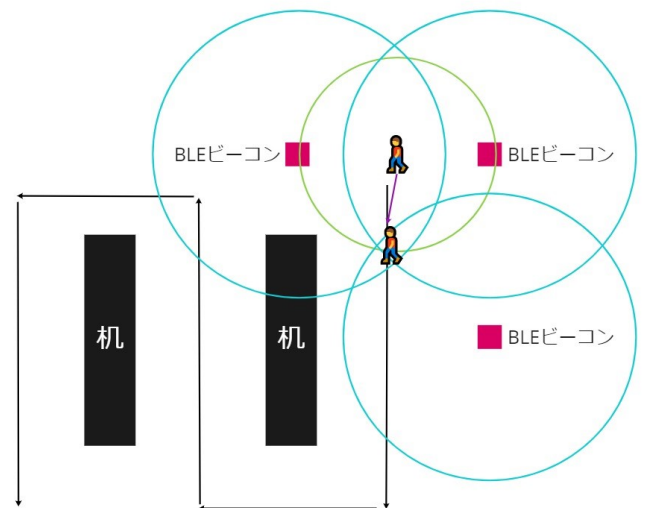


図 22 実験環境

図 22 では、実験環境を示している。図中の赤い点は BLE ビーコンを示し、BLE で 3 点測位を行い、加速度センサーで補正する。ユーザは机の間を逆 S 字に沿って歩行し、測定を行なっている。

5. 評価手法と分析手法

この章では実験で得られた結果の評価と分析をする。既存の手法と比べて精度評価をする。評価をするにあたり以下のデータを使用する。BLE だけで行う実験と BLE を加速度センサーで補正を行う実験をする。これらの 2 つの実験では、以下の 3 項目を比較する。

- 真っ直ぐコース通りに動いた場合
- 前後左右に動いた場合
- ぐちゃぐちゃに動いた場合

機械的に動く場合と、ユースケースを想定した場合で、移動距離と時間の誤差はどの程度変化しているのかを評価

する。さらに、BLE だけの誤差と加速度で補正する BLE の誤差によってどこまで解消されるのかを評価する。これは、距離が長くなると誤差がどの程度変化していくのかを評価する。

6. 議論

提案するシステムは 3 つのビーコンと加速度センサーで得られた歩幅の距離で補正を行い、精度改善する手法である。この提案では加速度の精度が高いほど精度が改善され、障害物や設置物のない広い広場で有効的に使用できる。しかし、本研究の加速度センサーの精度では直進した場合の距離しか測定ができない。そのため、1 歩だけではなく複数歩歩いた場合に誤差が大きくなる。そこでどちらに向けたかを判定できるセンサーを実装できれば、より精度改善に繋がると考えられる。

7. おわりに

位置情報を使用する IoT デバイスが増加してる中、屋内では GPS が届かない場合がある。屋内でも使用できる、ESP32 と BMX055 を使用し、BLE と加速度センサーで測位を行う。BLE だけでは誤差が大きいため、加速度を使って歩数を測定し、歩幅を距離変換し、歩幅と 2 つの RSSI の値と 3 点測位を行う。本研究の結果、RSSI の誤差より加速度の 1 歩の誤差が少ないことがわかった。そのため、歩幅の距離を 1 つのビーコンに置き換えて 3 点測位をすることによって、誤差が小さくなると考えられる。以上のことからビーコンの設置する量を減らして設置の場所を限定せず設置できるようになる。そのため、屋内の多くのビーコンを置けない場所で貢献できる。

参考文献

- [1] Ensworth, J. F. and Reynolds, M. S.: BLE-backscatter: Ultralow-power IoT nodes compatible with Bluetooth 4.0 low energy (BLE) smartphones and tablets, *IEEE Transactions on Microwave Theory and Techniques*, Vol. 65, No. 9, pp. 3360–3368 (2017).
- [2] Deng, Z., Yu, Y., Yuan, X., Wan, N. and Yang, L.: Situation and development tendency of indoor positioning, *China Communications*, Vol. 10, No. 3, pp. 42–55 (2013).
- [3] Yasir, M., Ho, S.-W. and Vellambi, B. N.: Indoor positioning system using visible light and accelerometer, *Journal of Lightwave Technology*, Vol. 32, No. 19, pp. 3306–3316 (2014).
- [4] Zolfaghari, A., El Said, M., Youssef, M., Zhang, G., Liu, T. T., Cattivelli, F., Syllaios, Y. I., Khan, F., Fang, F. Q., Wang, J. et al.: A multi-mode WPAN (Bluetooth, BLE, IEEE 802.15. 4) SoC for low-power and IoT applications, *2017 Symposium on VLSI Circuits*, IEEE, pp. C74–C75 (2017).
- [5] Ji, M., Kim, J., Jeon, J. and Cho, Y.: Analysis of positioning accuracy corresponding to the number of BLE beacons in indoor positioning system, *2015 17th International Conference on Advanced Communication*

- Technology (ICACT)*, IEEE, pp. 92–95 (2015).
- [6] Lee, M. S., Ju, H. and Park, C. G.: Map assisted PDR/Wi-Fi fusion for indoor positioning using smartphone, *International Journal of Control, Automation, and Systems*, Vol. 15, No. 2, p. 627 (2017).
- [7] Oguejiofor, O., Aniedu, A., Ejiolor, H. and Okolibe, A.: Trilateration based localization algorithm for wireless sensor network, *International Journal of Science and Modern Engineering (IJISME)*, Vol. 1, No. 10, pp. 2319–6386 (2013).
- [8] 太田 康平, 串田 高幸: BLE ビーコンを用いた移動型 IoT デバイスの位置情報の精度改善 (2021).
- [9] Maxwell Donelan, J., Kram, R. and Arthur D, K.: Mechanical and metabolic determinants of the preferred step width in human walking, *Proceedings of the Royal Society of London. Series B: Biological Sciences*, Vol. 268, No. 1480, pp. 1985–1992 (2001).
- [10] Hsu, C.-C., Chen, H.-C. and Lai, C.-Y.: An improved ultrasonic-based localization using reflection method, *2009 International Asia Conference on Informatics in Control, Automation and Robotics*, IEEE, pp. 437–440 (2009).
- [11] Chen, W., Chen, R., Chen, Y., Kuusniemi, H. and Wang, J.: An effective pedestrian dead reckoning algorithm using a unified heading error model, *IEEE/ION Position, Location and Navigation Symposium*, IEEE, pp. 340–347 (2010).
- [12] Pu, Y.-C. and You, P.-C.: Indoor positioning system based on BLE location fingerprinting with classification approach, *Applied Mathematical Modelling*, Vol. 62, pp. 654–663 (2018).