

平成 28 年度

卒業論文

近接情報を利用した屋内位置推定手法

千葉大学 工学部都市環境システム学科

松本 崇斗

指導教員： 塩田 茂雄 教授

目次

第1章	序論	1
1.1	背景	1
1.2	目的	1
1.3	本文書の構成	1
第2章	位置推定技術	2
2.1	レンジベース位置推定技術	2
2.1.1	TOA・TDOA	2
2.1.2	RSS	3
2.1.3	AOA	3
2.2	レンジフリー位置推定技術	4
2.2.1	Centroid法	4
2.2.2	DV-hop法	6
2.2.3	APIT法	7
第3章	無線規格	9
3.1	WLAN	9
3.2	ZigBee	11
3.3	UWB	12
3.4	Bluetooth	13
第4章	提案手法の説明	15
4.1	前提条件	15
4.2	位置推定手順	15
4.2.1	Stress Majorization	16
4.2.2	Coordinate System Registration	17
4.3	シミュレータの評価法	19

第 5 章 シミュレーション結果	21
5.1 シミュレーション条件 1	21
5.2 提案手法の精度評価	21
5.3 位置参照点数依存性	22
5.4 推定ユーザ数依存性	27
5.5 接続距離依存性	28
5.6 シミュレーション条件 2	30
第 6 章 結論	33
謝辞	34
参考文献	35

目次

2.1	TOA/TDOA/RSS を利用した位置検出技術	3
2.2	AOA を利用した位置検出技術	4
2.3	Centroid 法位置推定例	5
2.4	DV-hop 法位置推定例	7
2.5	APIT 法位置推定例	8
3.1	無線測位の仕組み	10
3.2	Zigbee 無線端末	11
3.3	IR-UWB リアルタイム測位システムの構成例	12
3.4	iBeacon の構成要素	13
3.5	iBeacon の測距	14
4.1	Coordinate System Registration の例	20
5.1	実際の位置と推定位置の比較	22
5.2	位置参照点数と誤差の関係	23
5.3	参照点数に関係した実際の位置と推定位置の比較	24
5.4	参照点数に関係した実際の位置と推定位置の比較 (続き)	25
5.5	参照点数に関係した実際の位置と推定位置の比較 (続き)	26
5.6	接続距離と推定ユーザ数の関係	27
5.7	接続距離と推定ユーザ数の関係 (続き)	28
5.8	接続距離と誤差の関係	29
5.9	実際の位置と推定位置の比較	30
5.10	位置参照点数と誤差の関係 (条件 2)	31
5.11	推定ユーザ数と誤差の関係 (条件 2)	31
5.12	接続距離と誤差の関係 (条件 2)	32

表 目 次

2.1	レンジベース位置推定技術の特徴	2
2.2	レンジフリー位置推定技術の特徴	4
3.1	無線規格の特徴	9
5.1	推定誤差	22

第1章 序論

1.1 背景

ナビゲーションシステムやセキュリティ、物流管理など様々な分野で位置情報を利用することが多い現在、グローバル・ポジショニング・システム（GPS）を代表とする位置推定技術が注目されている。GPSは衛星電波を利用した技術である。屋内や地下施設では衛星電波が届きにくく、GPSの利用ができない。そのため、屋内で位置推定する場合、GPSに代わる新たな位置推定手法が必要である。Wi-Fi, ZigBee, UWB (Ultra-wide Band), Bluetoothなどの電波（ビーコン）を発する装置を屋内に配置し、上記の装置からの電波をユーザが所持する端末で受信して位置を推定する方式が検討されている。

1.2 目的

一般に屋内では、位置情報は物品管理や動線管理に利用され、高い推測精度が要求される。本研究では、位置推定ユーザ間の近接情報（誰と誰が近い位置にいるか）を用いて、既存の屋内位置推定法の位置推定精度を向上させる手法を提案する。

1.3 本文書の構成

第1章では位置推定手法の背景と本論文の目的について書いた。第2章では既存の位置推定技術について触れた。第3章で、位置推定に利用される無線規格について述べた。第4章では、提案手法の条件と位置推定手順を解説した。第5章はシミュレーションの条件と結果、考察である。第6章で本論文の結論を述べた。

第2章 位置推定技術

位置推定技術には様々な手法があるが、大きくレンジベース方式とレンジフリー方式に大別される。以下では、位置推定環境に電波を発する端末を設置し、位置が未知のノードの位置推定を行う手法を紹介する。

2.1 レンジベース位置推定技術

レンジベース方式は各ノード間距離や受信信号の到来方向を測定結果から求めそれに基づきノードの位置を推定する手法である。TOA(Time Of Arrival), TDOA(Time Difference Of Arrival), RSS(Received Signal Strength) によりノード間距離を推定し位置推定を行う手法と、電波の到達角度によりノードの位置推定を行う AOA(Angle Of Arrival) という手法がある。レンジベース位置推定技術の特徴を表 2.1 に示す。

表 2.1: レンジベース位置推定技術の特徴

推定手法	推定要素	必要ノード数	問題点
TOA・TDOA	到達時刻	3	ノードの時間同期誤差が精度に影響
RSS	信号強度	3	距離減衰のモデル化が必要
AOA	到達角度	2	アンテナ設置のコスト

2.1.1 TOA・TDOA

到達時間 (TOA : Time Of Arrival) 方式はノード間の電波送信時間と受信時間を測定することにより到達距離を求める方式である。そのため、送信側と受信側の時間同期が必要となる。同期時の時間誤差が推定の精度に影響する。

到達時間差 (TDOA : Time Difference Of Arrival) 方式は、各電波受信側での電波の到達時間差により位置を推定する手法である。送信側との同期をする必要がないため、TOA 方

式よりも実用的である。

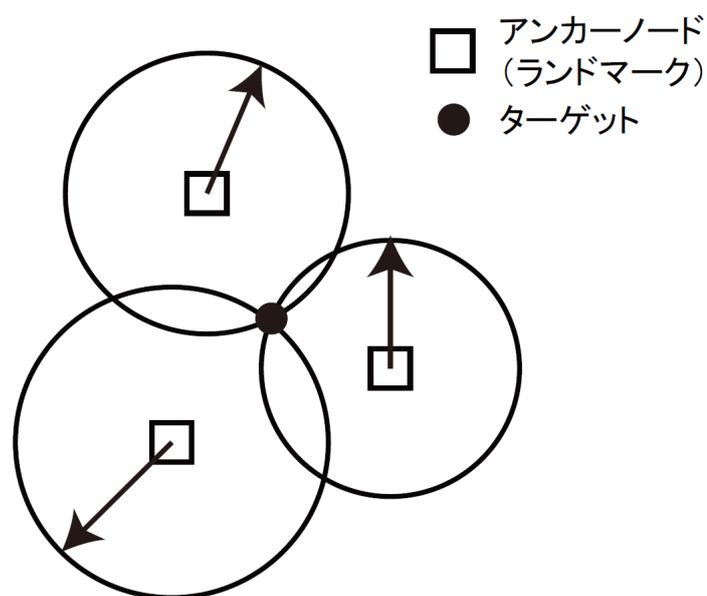


図 2.1: TOA/TDOA/RSS を利用した位置検出技術
出典) [1][2]

2.1.2 RSS

RSS 位置検出技術は、ノードが発した電波を受信し、その受信信号強度から距離を算出、位置を推定する手法である。受信信号強度から距離を算出するには、その環境における電波の距離減衰のモデル化が必要である。距離減衰のモデル化には使用環境下で様々な位置、距離の RSS の測定が必要である。簡易な装置で電波を受信、位置を算出できるが、推測の精度を向上するためには、距離減衰を複数モデル化する必要がある。

2.1.3 AOA

到達角度 (AOA : Angle Of Arrival) 方式は、アンカーノードが発した電波がどの方向 (到達角度) から発せられたかを測定することで位置を推測する。図に AOA 方式の推定手法を示す。AOA 方式が必要とする情報は到達角度 θ_1 と θ_2 のみであるため、TOA・TDOA や RSS に比べ測位するためのノード数が少ない利点がある。しかし、到達角度を推定するには、指向性のアンテナやアレーアンテナを用いるため、コストがかかってしまう欠点がある。

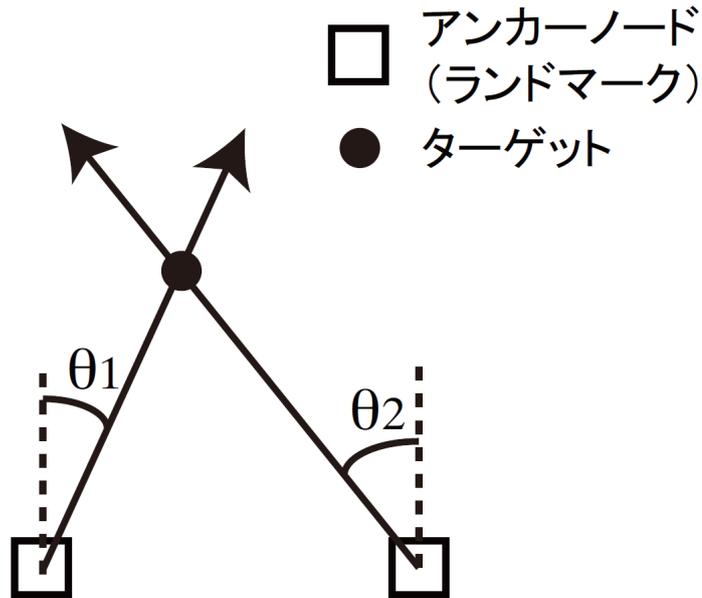


図 2.2: AOA を利用した位置検出技術
出典) [1][2]

2.2 レンジフリー位置推定技術

レンジフリー方式はあらかじめ位置がわかっている一部のノード（アンカーノード）を用い、それらノードとの位置関係やホップ数などからノードの位置を推定する手法である。レンジフリー位置推定技術の特徴を表 2.2 に示す。

表 2.2: レンジフリー位置推定技術の特徴

推定手法	推定要素	必要ノード数	問題点
Centroid	重心位置	3	アンカーノード設置のコスト
DV-hop	平均距離	3	誤差が大きい
APIT	APIT 検定と RSSI	3	RSSI 誤差が精度に影響

2.2.1 Centroid 法

Centroid 法 [3] は、アンカーノードが発したビーコンをノードが取得することで、位置推定をする。各ノードは、ビーコンからアンカーノードの位置を受信する。複数のアンカーノードの位置の重心をノードが存在する位置として算出することで位置推定を行う。式に N 台のアンカーノード位置 $(X_1, Y_1) \cdots (X_N, Y_N)$ をノードが取得したとき、ノードの位置とし

て推測される座標 (X, Y) を示す.

$$(X, Y) = \left(\frac{X_1 + \dots + X_N}{N}, \frac{Y_1 + \dots + Y_N}{N} \right) \quad (2.1)$$

図 2.3 にセントロイドアルゴリズムでノード位置を推定する例を示す. アンカーノードを青丸, 推定ノードを赤丸で示す. アンカーノードが発するビーコンの到達距離を黒丸で表した.

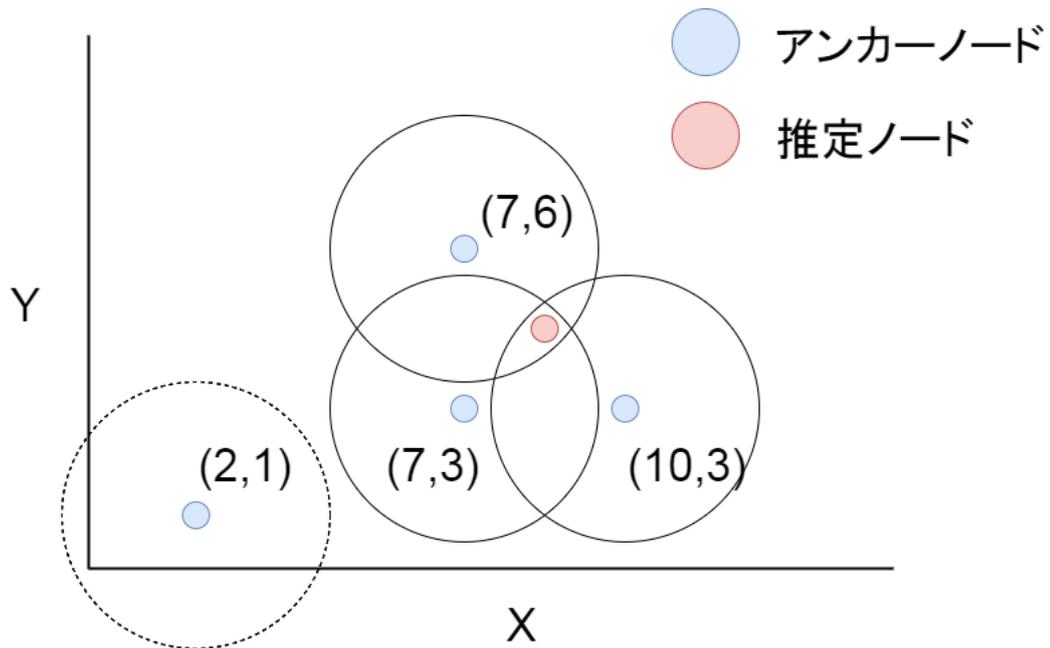


図 2.3: Centroid 法位置推定例

上記の例ではアンカーノードの座標はそれぞれ, $(7, 3), (7, 6), (10, 3)$ であるので

$$(8, 4) = \left(\frac{7 + 7 + 10}{3}, \frac{3 + 6 + 3}{3} \right) \quad (2.2)$$

となり, 推定ノードの位置は $(8, 4)$ だと推定される.

Centroid 法は手法自体は簡易ではあるが, 位置推定の精度を上げるためには, 各端末を多くのアンカーノードがカバーする必要がある. そのため, Centroid 法にはコストや設置場所の増加といった問題点がある.

2.2.2 DV-hop 法

DV-hop 法 [4] は、アンカーノードからノードへのホップ数を取得後、1 ホップの平均距離よりノードへの距離を計算し、位置を推定する手法である。以下に手順を示す。

1. 各ノードがアンカーノードから電波を受信する
2. 受信データからノードはアンカーノードとノード間のホップ数を取得する
3. アンカーノード間距離より 1 ホップの平均距離を算出する
4. 全てのアンカーノード間において、手順 3 を行う。
5. アンカーノードが 1 ホップの平均距離を近隣のノードへ送信する
6. ノードがアンカーノードとの距離を算出する
7. 以上の手順で取得したデータより多角測量し、位置推定する

手順 6 では、3 個以上のアンカーノードとの距離を算出する必要がある。手順 3 は式が用いられる。 C_i はアンカーノード i に関する 1 ホップの平均距離である。

$$C_i = \frac{\sum \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}}{\sum h_i} \quad (i \neq j) \quad (2.3)$$

上記の例では C_1, C_2, C_3 [m] はアンカーノード i から推定ノードまでの 1 ホップの平均距離を表し、これらを計算すると

$$C_1 = \frac{70 + 100}{2 + 4} = 28.3 \quad (2.4)$$

$$C_2 = \frac{70 + 120}{2 + 4} = 31.7 \quad (2.5)$$

$$C_3 = \frac{100 + 120}{4 + 2} = 36.7 \quad (2.6)$$

である。アンカーノード A1, A2, A3 の推定ノードまでのホップ数はそれぞれ 2, 3, 2 であるので、ホップ数を 1 ホップの平均距離に乘じ算出すると、アンカーノードから推定ノードまでの距離 L_i [m] は

$$L_1 = 2 \times 28.3 = 56.6 \quad (2.7)$$

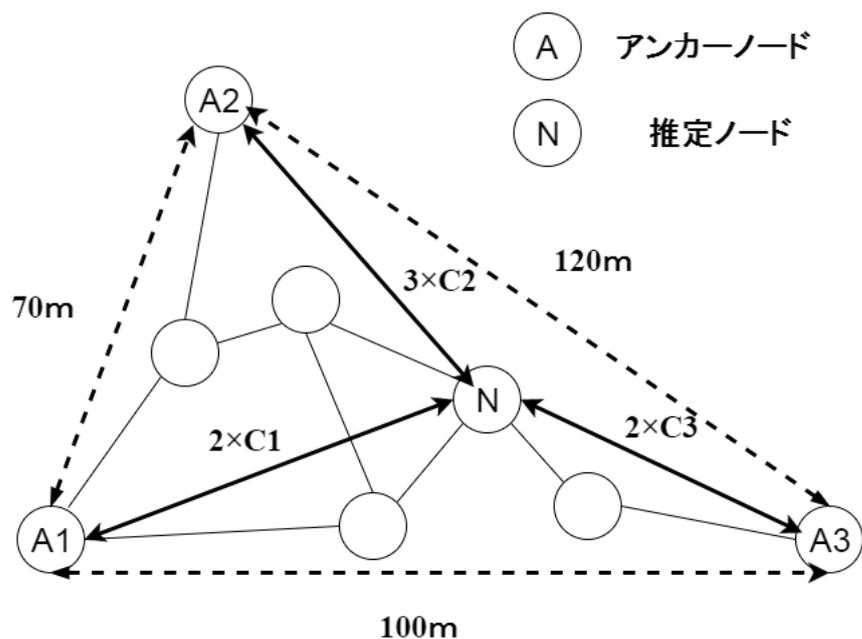


図 2.4: DV-hop 法位置推定例

$$L_2 = 3 \times 31.7 = 95.1 \tag{2.8}$$

$$L_3 = 2 \times 36.7 = 110.1 \tag{2.9}$$

となる。アンカーノードの位置が分かっているので、多角測量に基づき、位置を推定する。DV-hop 法の手法は簡単であるが、誤差が大きい。また、アンカーノードとノードの数が増えると、通信量が増大する問題がある。

2.2.3 APIT 法

APIT (Approximate Point-In-Triangulation Test) 法 [5] は、各アンカーノードがそれ自身の位置情報を含んだビーコンを定期的にブロードキャストする。各ノードは、受信したビーコンから 3 つのアンカーノードの組み合わせで作成可能なすべての三角形を求める。これらの三角形に対し、自分が各三角形の内側にいるのか外側にいるのかを検証し、それに基づき自身の位置を絞り込んでいく。三角形の内部か外部の判断には、PIT 検定および APIT 検定が用いられる。PIT 検定は、各ノードをすべての方角に移動し、各三角形の 3 台すべてのノードとの距離が離れる方角がある場合、そのノードはその三角形の外側にいると判定する。簡単な検定法ではあるが、各ノードをすべての方角に移動するのは現実的ではない。

これを解決する方法として、ノードを移動させるのではなく、RSSI を利用するのが APIT 検定である [6].

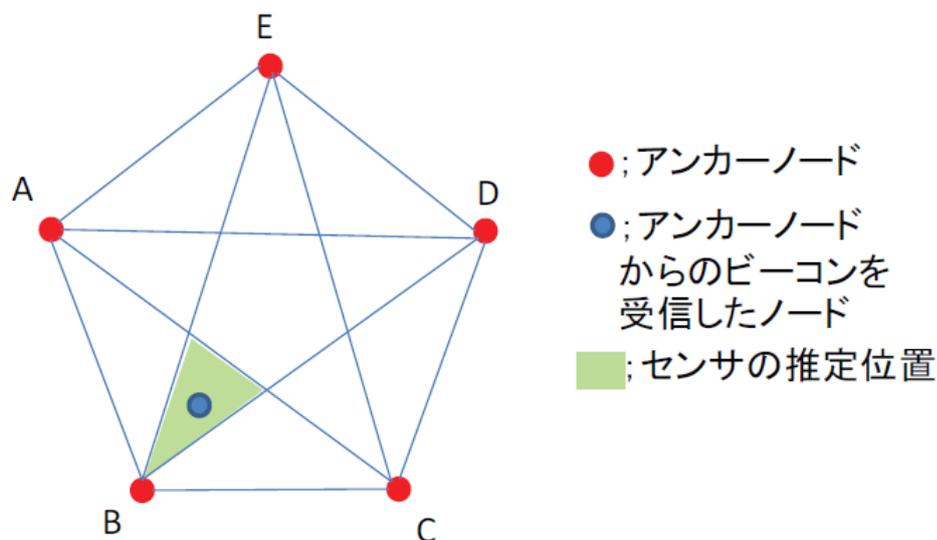


図 2.5: APIT 法位置推定例
出典) [6]

APIT 検定では、各ノードは各アンカーノードからの RSSI 情報を交換し、各アンカーノードに対し、どちらのノードの方が近いかをまず決める。その情報に基づき、各三角形を構成する 3 つのアンカーノードとのすべての距離が、自分より遠いノードが近隣ノードに存在する場合、自分がその三角形の外側にいると判定する。APIT 法は、少ない通信量（オーバーヘッド）で Centroid 法よりも高い位置推定精度を得られるが、アンカーノードで構成されるどの三角形の内側にも含まれないノードの位置を決めることはできない。また、RSSI と距離との関係を表す減衰モデルは近似モデルであり、RSSI から求められる距離は必ずしも正確ではない。そのため、三角形の内側にいるか外側にいるかの判定は誤りやすく、位置推定精度も劣化してしまう [6].

図に APIT 法での位置推定例を示す。推定ノードは A から E, 5 個のアンカーノードからビーコンを受信する。その後、5 個のアンカーノードから、3 個のアンカーノードを組み合わせる。次に RSSI により距離を測定し、推定ノードがどの三角形に含まれているかを推測する。図では推定ノードは三角形 ABC, BCE, BDE に含まれる。そのため、推定ノードは三角形 ABC, BCE, BDE が重なっている位置に存在すると推定できる。

第3章 無線規格

位置推定には無線 LAN, ZigBee, UWB, Bluetooth などの無線規格が用いられる。上記の無線規格の特徴を表 3.1 に示す。

表 3.1: 無線規格の特徴

無線規格	最大転送速度 (bps)	最大転送距離 (m)	消費電力 (mW)	同時接続可能数
WLAN	54M	約 100	1000	32
ZigBee	250k	約 70	60	65535
UWB	480M	約 10	100	256
Bluetooth	24M	約 100	120	7

3.1 WLAN

WLAN(Wireless Local Area Network) とは無線通信を利用し、データの送受信を行う LAN システムのことである。転送速度が高速であり、転送可能距離も長い。WLAN 規格のひとつである Wi-Fi 規格を持つ端末による位置推定が普及している。Wi-Fi 測位は既設のアクセスポイントを利用できるため、設備投資を抑えられるメリットがある。また、スマートフォンに Wi-Fi 機能が標準搭載されているため、ユーザが Wi-Fi 位置推定手法を利用しやすい。今主流となっている位置推定方法が GPS とこの Wi-Fi 測位を組み合わせたハイブリッド型である。Wi-Fi を利用する位置推定では、Place Engine がある [7]。Place Engine は近傍の WiFi アクセスポイントから電測情報として電波を観測し、Place Engine サーバへ送信する。そして、送信された情報とサーバに蓄えられている WiFi 電測情報のデータベースから位置情報を推定し、クライアントに返す (Location Fingerprint 法)。しかし、WiFi 電測情報のデータベースの状況により、位置推定精度の正確さを測るのは難しく、5m ~100m 程度の範囲で位置が推定される。

以下に Wi-Fi 測位の具体的な手順を示す。

1. 端末が Wi-Fi アクセスポイントから電波を受信する
2. 電波からアクセスポイントの SSID, MAC アドレス等を取得
3. 端末が SSID, MAC アドレス等を元に Place Engine サーバへ問い合わせ
4. Place Engine データベースからおおまかな位置情報を取得
5. RSSI によりアクセスポイントからの距離を算出, 位置推定する

手順 5 では図 3.1 に示した Wi-Fi の受信信号の強度と時間から距離を測定したグラフによって, アクセスポイントの距離を算出する。

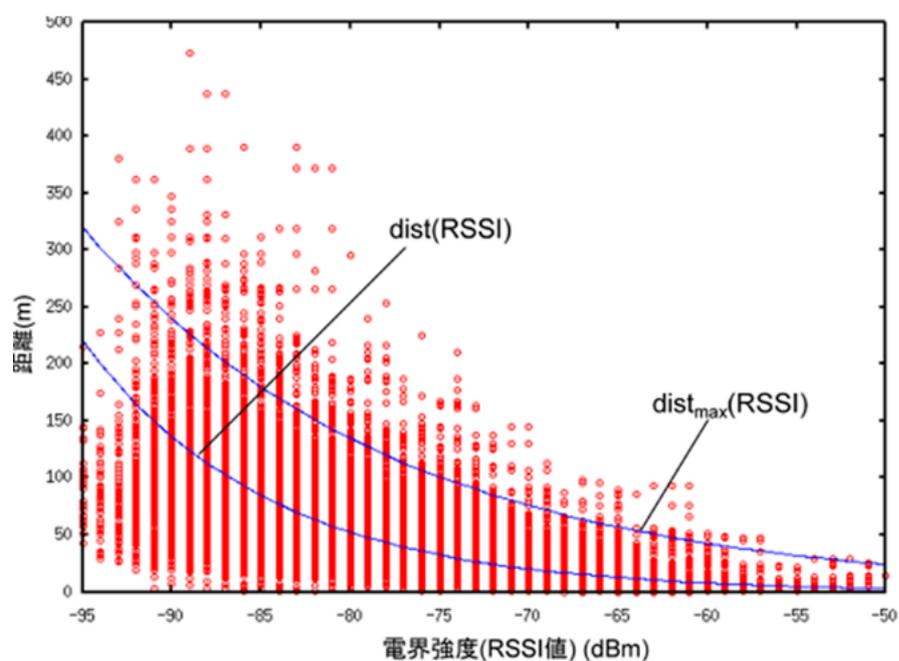


図 3.1: 無線測位の仕組み
出典) [7]

3.2 ZigBee

ZigBee は近距離無線通信規格のひとつである。転送速度が低速である代わりに、消費電力が少ない。使用周波数は 2.4GHz 帯である。ZigBee 規格の端末を図に示す。図 3.2 は Xbee と呼ばれるモジュールで 25mm × 28mm と小型、安価であり、マイコンに組み込むことができる。Xbee はモデルにもよるが、送信電流 45mA、受信電流 50mA で、待機時のパワーダウン電流は 10 μ A 以下と極めて消費電力が低く、ボタン電池 1 個でおよそ 1 年間の稼働が可能である。また、短時間 (15ms) で、待機時から復帰時への復帰が可能である。リアルタイムでの位置測定には向かないが、アーバンレンジ（電波を阻害するものが多い状態における通信距離）が約 60m であり、屋内の位置推定に実用的であると言える。同時接続可能数が多いことを活かして、メッシュネットワークを形成できるため、本提案手法と相性が良いと考察される。



図 3.2: Zigbee 無線端末

3.3 UWB

超広帯域無線通信 (UWB : Ultra Wide Band) とはその名の通り、数 GHz に渡る広い帯域を利用する通信方式である [9]。UWB を利用した測位方法は 3 個以上の固定機と UWB 移動機との距離を到達時間により計測して位置を推定する (TOA・TDOA 方式)。NICT が開発中の屋内測位システムでは、インパルス方式 UWB (IR-UWB) を採用している。IR-UWB は非常に短いパルス幅の電波を発生し、数 Gbps の高速通信が可能である。そのため、リアルタイムで誤差数 10cm の高い精度の位置推定ができる。しかし、UWB は最大転送距離が短く、位置推定のための固定機を多数設置する必要があるため、専用の移動機も必要となることから、コストの面で実用化が難しいとされている。

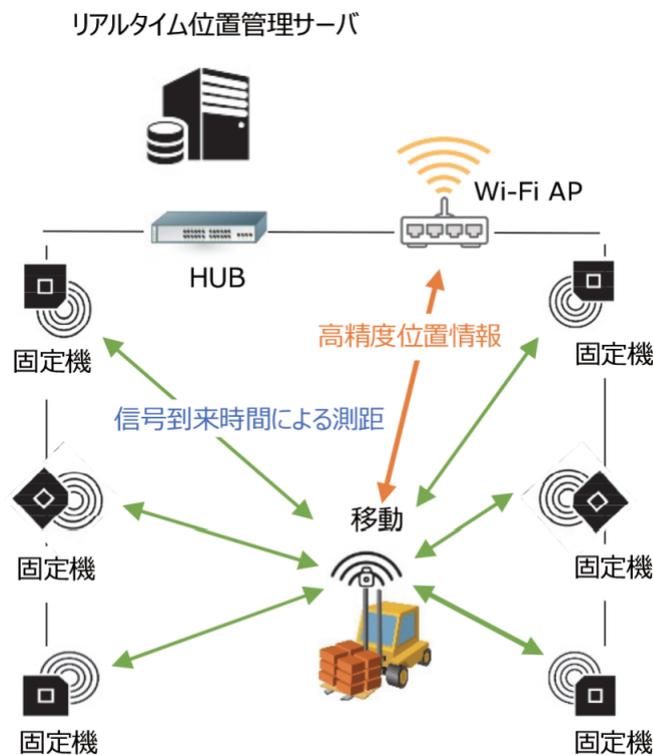


図 3.3: IR-UWB リアルタイム測位システムの構成例 (出典) [8]

3.4 Bluetooth

Bluetooth とは近距離無線通信規格のひとつである。消費電力が少なく高速通信が可能である。Bluetooth3.0 以後のバージョンに、BLE(Bluetooth Low Energy) という規格があり、省電力かつ省コストで通信が可能である [10].

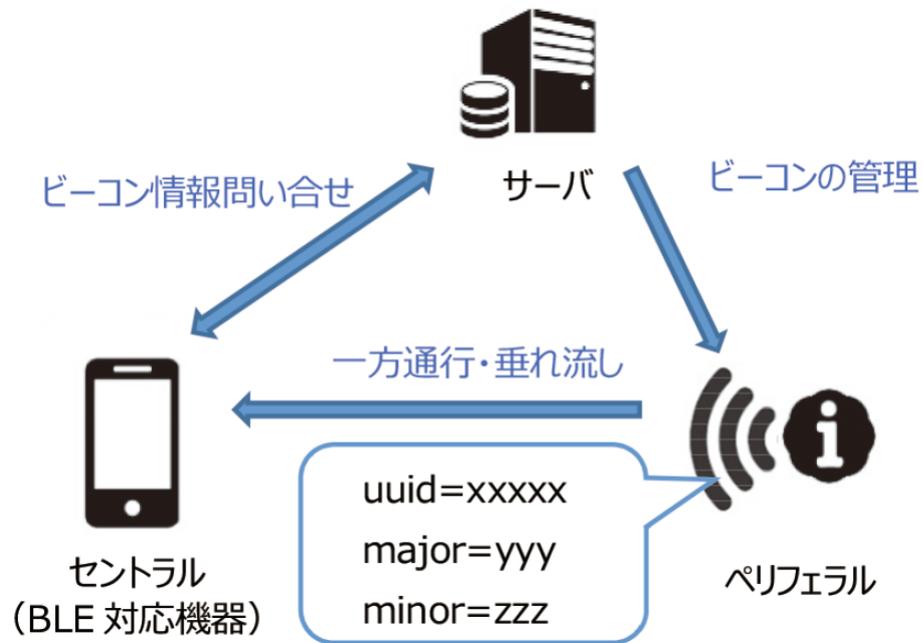


図 3.4: iBeacon の構成要素
出典) [8]

現在 BLE モジュールによる屋内位置推定技術 (iBeacon) が、店舗内の消費者の動向調査や店内の商品案内、クーポン発行などに利用されている。iBeacon ではビーコンを発信する機器を「ペリフェラル」、スマートフォンを一例とする、ビーコンを受信する機器をセントラルと呼ぶ。ペリフェラルは一方的にビーコンをセントラルに対し送信するのみで、相互通信は行われない。そのため、iBeacon を用いた測位では、セントラルと通信を行うサーバが必要である。

iBeacon を用いた測位手法ではセントラルとペリフェラル間の距離の絶対的な距離は測位できないが、ペリフェラルが存在する領域への入出の検知と、セントラルとペリフェラル間の距離が「至近」、「近い」、「遠い」、「不明」といった相対的な距離が検出できる。本論文で紹介する手法は近接情報を利用するので、絶対距離の取得ができなくても相対距離の取得ができれば問題がない。そのため、省コストで相対的な距離が検出できる iBeacon との相性が良いと考察される。

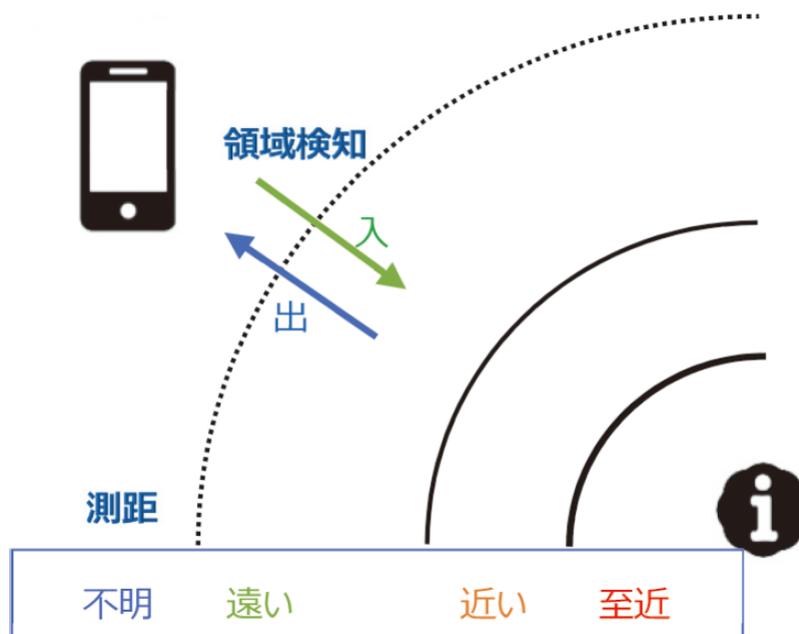


図 3.5: iBeacon の測距
出典) [8]

第4章 提案手法の説明

本章では、提案手法の前提条件と具体的な手順を解説する。

4.1 前提条件

位置推定ユーザと、(Wi-Fi アクセスポイント, RFID など) 電波を発する位置参照点が存在する屋内や地下施設を考える。以下では、位置推定ユーザと位置参照点を区別せず、単に「ユーザ」と呼ぶこととする（位置参照点は位置が既知のユーザとみなす）。ユーザには一連の番号を振り、 i 番目のユーザの推定位置（位置参照点の場合は実際の位置）を \mathbf{x}_i で表す。位置参照点に加えて位置推定ユーザも（Wi-Fi などの）信号を発する端末を所持し、それぞれのユーザが発する信号に基づき、各ユーザはどのユーザが近くにいるかを知ることができる。以下では、近い位置にいるユーザ同士は「接続している」と呼ぶこととする。

4.2 位置推定手順

位置推定は以下の手順で行う。

1. 近接情報に基づき、ユーザ間の相互位置関係を表す相対位置マップを生成する。
2. 位置参照点の情報を用いて、相対位置マップに並進、回転、尺度変換を施し、絶対位置マップを生成し、位置推定値を出力する。

ユーザ間の近接情報を用いると、接続しているユーザ間をリンクで結んだユーザの「グラフ」が構成できる。手順1は、このグラフ上でのユーザ $i - j$ 間のホップ数を h_{ij} として、

$$\sum_{ij} (|\mathbf{x}_i - \mathbf{x}_j| - h_{ij})^2 \quad (4.1)$$

を最小化する相対位置マップを生成するものであり，グラフ描画の手法として使われている Stress Majorization[11] を適用する．

手順 2 は，手順 1 で生成した相対位置マップ上の位置参照点の位置が，実際の位置参照点の位置になるべく近づくように，相対位置マップの並進，回転，尺度変換を施して，絶対位置マップを生成するものであり，[12] で提案されている Coordinate System Registration を適用する．

本実験では 2 次元平面上のノード位置の推定を行う．

4.2.1 Stress Majorization

Stress Majorization はグラフの仮想的な物理モデルにもとづいて定義されたストレス関数を最小化することによって最適なグラフを描画する手法である．

h_{ij} はノード i とノード j について与えられた理想距離（ホップ数）である．ノード i が座標 \mathbf{x}_i にあるときストレス関数は

$$\sum_{ij} (|\mathbf{x}_i - \mathbf{x}_j| - h_{ij})^2 \quad (4.2)$$

である．すなわち，ストレス関数の最小化は， $(|\mathbf{x}_i - \mathbf{x}_j| - h_{ij})^2$ の和が最小化する，相対座標グラフ上での各ノードの推定座標 $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ の組み合わせを見つけることと言い換えることができる．

$$\operatorname{argmin} \varepsilon(\mathbf{x}_1, \dots, \mathbf{x}_N) = \sum_{ij} (|\mathbf{x}_i - \mathbf{x}_j| - h_{ij})^2 \quad (4.3)$$

$(|\mathbf{x}_i - \mathbf{x}_j| - h_{ij})^2$ の和が最小化する，相対座標グラフ上での各ノードの推定座標 $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ の組み合わせは全てのノード間で $|\mathbf{x}_i - \mathbf{x}_j| = h_{ij}$ もしくは $|\mathbf{x}_i - \mathbf{x}_j| \simeq h_{ij}$ が満たされるため，センサの最適な相対座標を与えると期待できる．実際は，全てのノード間で $|\mathbf{x}_i - \mathbf{x}_j| = h_{ij}$ を満たす $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ の組み合わせは存在しないため，この事実が提案手法の推測誤差になる．また，最小値を求めるために，最急降下法を用いるが，局所的な最小値に陥ると，推測誤差の原因となる．これを避けるため，初期値を複数回与え，最もストレス関数が最小化されたときの $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ の組み合わせを相対座標上の各ノードの推定座標とした．

最急降下法により，ストレス関数の傾きから，ストレス関数の最小値を探索する．最急降下法では反復法を用いて関数を解に近づけていく． k 回目の反復のときノード i の座標を \mathbf{x}_i^k とすると， \mathbf{x}_i^{k+1} は

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k - \alpha \left. \frac{\partial \varepsilon}{\partial \mathbf{x}_i} \right|_{\mathbf{x}_i = \mathbf{x}_i^{k+1}} \quad (4.4)$$

と表せられる．本手法ではパラメータ α は 10^{-5} とした．

また， $\frac{\partial \varepsilon}{\partial \mathbf{x}_i}$ は

$$\begin{aligned} \frac{\partial \varepsilon}{\partial \mathbf{x}_i} &= \sum_{ij}^N \frac{\partial (|\mathbf{x}_i - \mathbf{x}_j| - h_{ij})^2}{\partial \mathbf{x}_i} \\ &= 2 \sum_{ij}^N (|\mathbf{x}_i - \mathbf{x}_j| - h_{ij}) \frac{\partial |\mathbf{x}_i - \mathbf{x}_j|}{\partial \mathbf{x}_i} \end{aligned} \quad (4.5)$$

となる．シミュレーションではこれを反復して，ストレス関数の最小値を求める．また，ストレス関数が最小のときの $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ を相対座標上の推定ノードとする．

4.2.2 Coordinate System Registration

Coordinate System Registration はアンカーノード（位置参照点）を基準として，グラフに座標変換を加え，相対座標グラフを絶対座標グラフにする手法である．

座標の変換は

$$\mathbf{x}' = sR\mathbf{x} + \mathbf{t} \quad (4.6)$$

に基づき，並進，回転，尺度変換のパラメータはそれぞれ t, R, s である．以下に Coordinate System Registration の手順を記述する．相対座標と絶対座標においてアンカーノード i の二次元座標をそれぞれ $\mathbf{x}_{l,i}$ と $\mathbf{x}_{r,i}$ とする．

Step1 相対座標と絶対座標のアンカーノードの重心を計算する．

$$\bar{\mathbf{x}}_l = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_{l,i} \quad \bar{\mathbf{x}}_r = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_{r,i} \quad (4.7)$$

ここで n はアンカーノードの数である.

Step2 重心に基づいてアンカーノードを移動する.

$$\mathbf{x}'_{l,i} = \mathbf{x}_{l,i} - \bar{\mathbf{x}}_l \qquad \mathbf{x}'_{r,i} = \mathbf{x}_{r,i} - \bar{\mathbf{x}}_r \qquad (4.8)$$

Step3 尺度変換のパラメータ s を計算する.

$$s = \sqrt{\frac{\sum_{i=1}^n \|\mathbf{x}'_{r,i}\|^2}{\sum_{i=1}^n \|\mathbf{x}'_{l,i}\|^2}} \qquad (4.9)$$

Step4 M を計算する.

$$M = \sum_{i=1}^n \mathbf{x}'_{r,i} (\mathbf{x}'_{l,i})^T \qquad (4.10)$$

Step5 $M^T M$ の固有値分解を計算する. $M^T M$ は対称行列であり, 固有値 λ と固有ベクトル $\hat{\mathbf{u}}$ は以下となる.

$$M^T M = \lambda_1 \hat{\mathbf{u}}_1 \hat{\mathbf{u}}_1^T + \lambda_2 \hat{\mathbf{u}}_2 \hat{\mathbf{u}}_2^T \qquad (4.11)$$

Step6 $S = (M^T M)^{1/2}$ より $R = MS^{-1}$ を計算する.

$$\begin{aligned} S &= (M^T M)^{1/2} \\ &= \sqrt{\lambda_1} \hat{\mathbf{u}}_1 \hat{\mathbf{u}}_1^T + \sqrt{\lambda_2} \hat{\mathbf{u}}_2 \hat{\mathbf{u}}_2^T \end{aligned} \qquad (4.12)$$

であるから

$$S^{-1} = \frac{1}{\sqrt{\lambda_1}} \hat{\mathbf{u}}_1 \hat{\mathbf{u}}_1^T + \frac{1}{\sqrt{\lambda_2}} \hat{\mathbf{u}}_2 \hat{\mathbf{u}}_2^T \qquad (4.13)$$

これを R に代入すると

$$\begin{aligned} R &= MS^{-1} \\ &= M \left(\frac{1}{\sqrt{\lambda_1}} \hat{\mathbf{u}}_1 \hat{\mathbf{u}}_1^T + \frac{1}{\sqrt{\lambda_2}} \hat{\mathbf{u}}_2 \hat{\mathbf{u}}_2^T \right) \end{aligned} \quad (4.14)$$

となり回転行列 R が求まる.

Step7 これまでのパラメータから並進変換のパラメータ t を計算する.

$$t = \bar{\mathbf{x}}_r - sR\bar{\mathbf{x}}_l \quad (4.15)$$

Step8 s , R , t よりノード i の推測座標 \mathbf{x}'_i を求める.

$$\mathbf{x}'_i = sR\mathbf{x}_i + t \quad (4.16)$$

Step8 を相対座標グラフのノード全てに適用することで, 絶対座標グラフが求まる.

図 4.1 に Coordinate System Registration の例を示す. 左上の図が \mathbf{x}_r の集合である. その下の図が, \mathbf{x}_l の集合が \mathbf{x}_r の基準点に従って, 新しい座標系に移動し, 位置誤差をシミュレートするようにジッターリングされている図である. 最後に右の図で \mathbf{x}_l 座標と \mathbf{x}_r 座標で位置が合わされる.

4.3 シミュレータの評価法

位置推定精度は測定誤差から平均推定誤差を算出し, 評価する. 測定誤差は全てのユーザ間の実際の座標 \mathbf{x}_i と, 推定座標 \mathbf{x}'_i の距離 (誤差) の平均である. 全ユーザ数を N としたとき, 推定誤差 $error[m]$ は式 5.1 で表される.

$$error = \frac{\sum_{i=1}^N |\mathbf{x}_i - \mathbf{x}'_i|}{N} \quad (4.17)$$

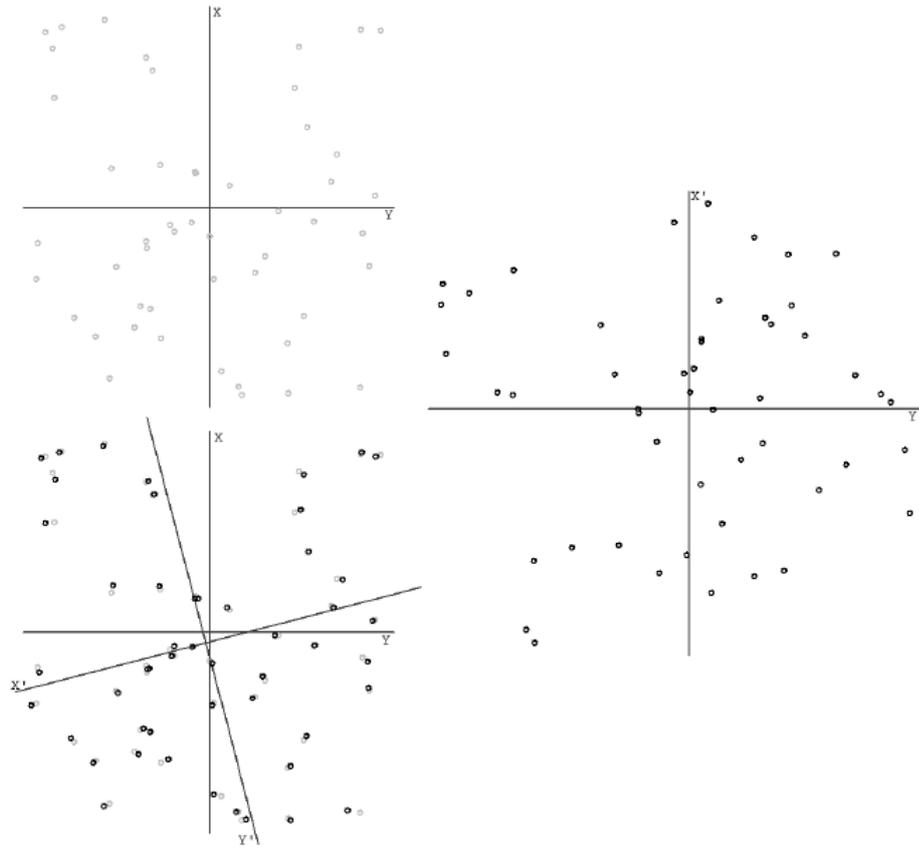


図 4.1: Coordinate System Registration の例 [12]

また、平均推定誤差 [m] は n 回目のシミュレーション時の推定誤差を $error_n$ とし、シミュレーション回数を L とすると

$$\text{平均推定誤差} = \frac{\sum_{n=1}^L error_n}{L} \quad (4.18)$$

である。

第5章 シミュレーション結果

5.1 シミュレーション条件1

下記の条件のもとシミュレータを作成し，位置推定シミュレーションした．ユーザをランダムに配置するプログラム，Stress Majorization を行うプログラム，Coordinate System Registration をするプログラムを C 言語で作成した．

- 30m 四方の部屋に 20 人の位置推定ユーザが存在
 - ユーザは部屋内にランダムに位置する
- 位置参照点を部屋内に 5 個設置
 - 部屋の四隅にそれぞれ 1 個設置
 - 部屋の中央に 1 個設置
- 距離 20m 以内のユーザ同士は「接続している」

図 5.1 に実際の位置と推定位置の比較を示した．赤点がユーザの実際の位置，青のバツ印がシミュレーションで推定された位置である．ユーザの推定誤差は青の破線で示す．良好な位置推定結果が得られている．

5.2 提案手法の精度評価

表 5.1 にセントロイドアルゴリズム（近い位置にある位置参照点の重心を位置推定値とする手法）と近接情報利用型位置推定法の精度の比較を示す．100 回シミュレーションを行い，推定誤差の平均を算出した．

セントロイドアルゴリズムの平均推定誤差は 4.12m であるが，提案手法の誤差はセントロイドアルゴリズムの誤差の半分程度の 2.45m であった．この結果から，提案手法では従来のアルゴリズムよりも，高い精度で推測することが可能であると言える．

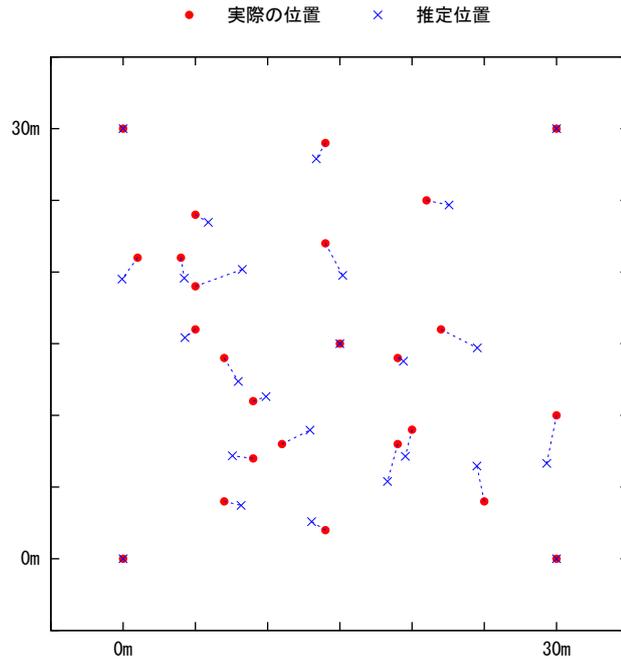


図 5.1: 実際の位置と推定位置の比較

表 5.1: 推定誤差

手法	平均推定誤差
提案手法	2.45 m
セントロイドアルゴリズム	4.12 m

5.3 位置参照点数依存性

位置参照点の数によって、精度に変化があるか調べた。図 5.2 に 5.1 と同様の条件のもと位置参照点を 4,9,16,25,36,49 個とし、配置したときの平均推測誤差を示す。位置参照点は座標面上に均等に配置した。また、セントロイドアルゴリズムについても同様に平均推測誤差を算出し図 5.2 に示した。

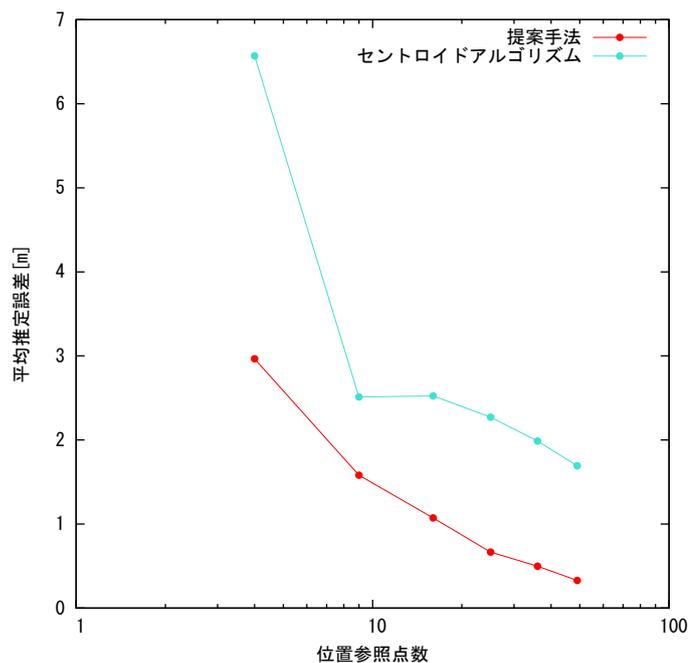
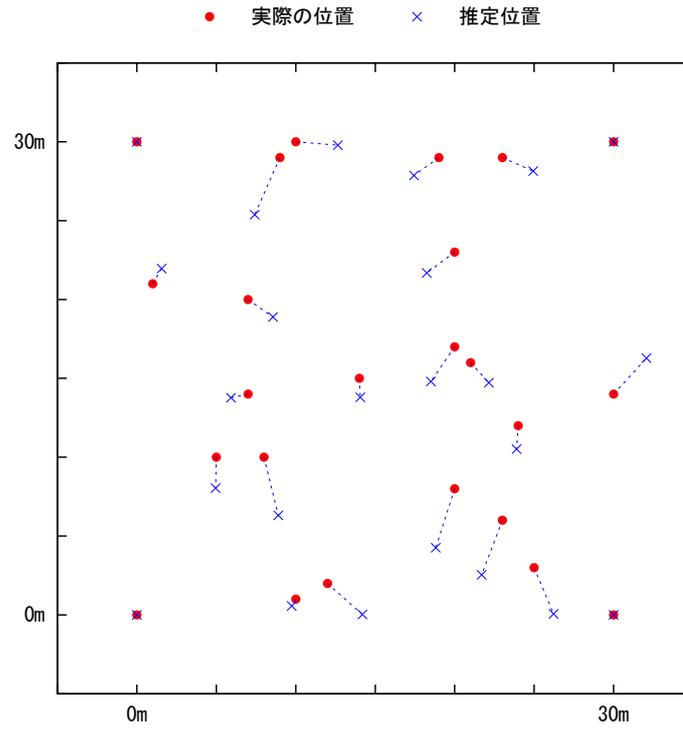
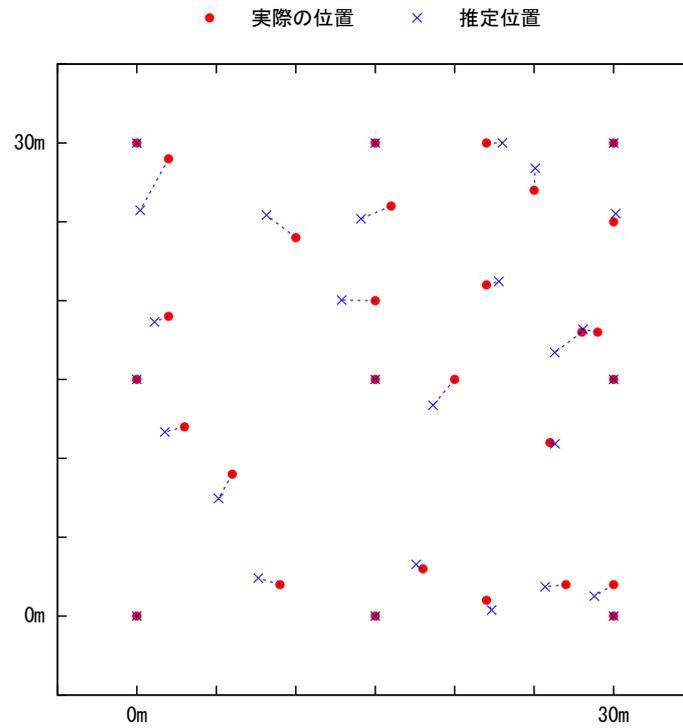


図 5.2: 位置参照点数と誤差の関係

提案手法の推定誤差が位置参照点数の 2 乗に比例して減少していることが分かる。位置参照点を多く配置すると、多くの近接情報が得られるためであると考えられる。また、参照点が増えると、[2] の手法を用いた座標変換の精度が上がるということが分かっている。そのため、相対位置マップ上の参照点が絶対位置マップ上の実際の位置により近づき、適切な絶対位置マップが描画され、推定誤差が減少したものと考察できる。セントロイドアルゴリズムも位置参照点数が増加すると、平均推定誤差が小さくなる。セントロイドアルゴリズムと提案手法を比較すると、どの位置参照点数においても、平均推定誤差が提案手法の方が小さい。この結果からも、提案手法では従来のアルゴリズムよりも、高い精度で推測することが可能であると言える。

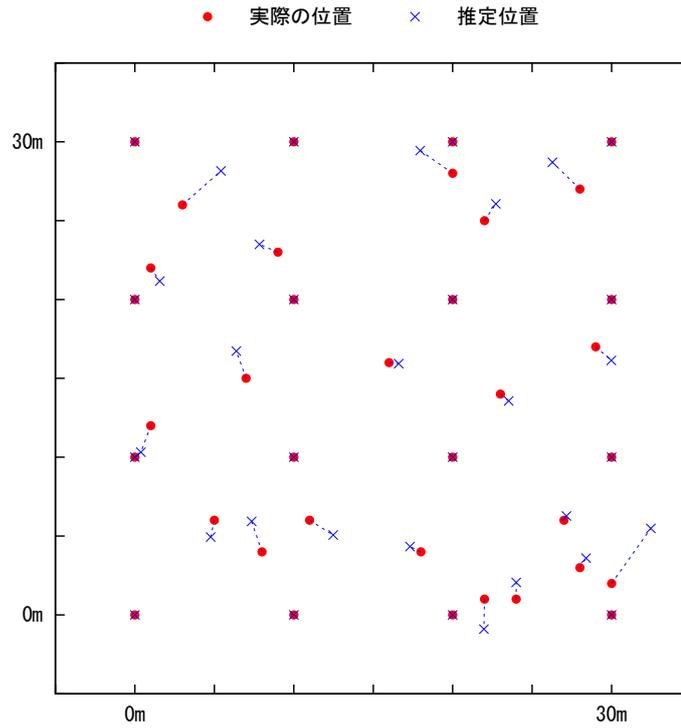


(a) 位置参照点 4 個

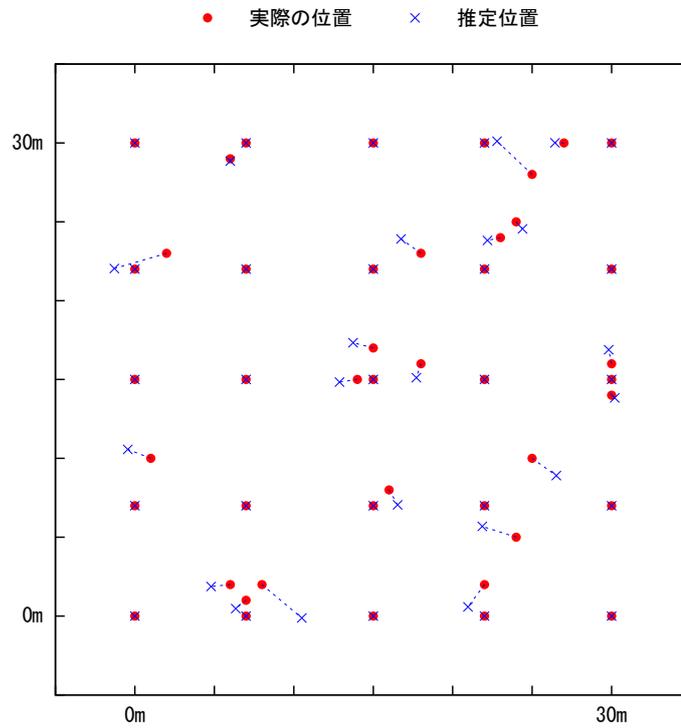


(b) 位置参照点 9 個

図 5.3: 参照点数に関係した実際の位置と推定位置の比較

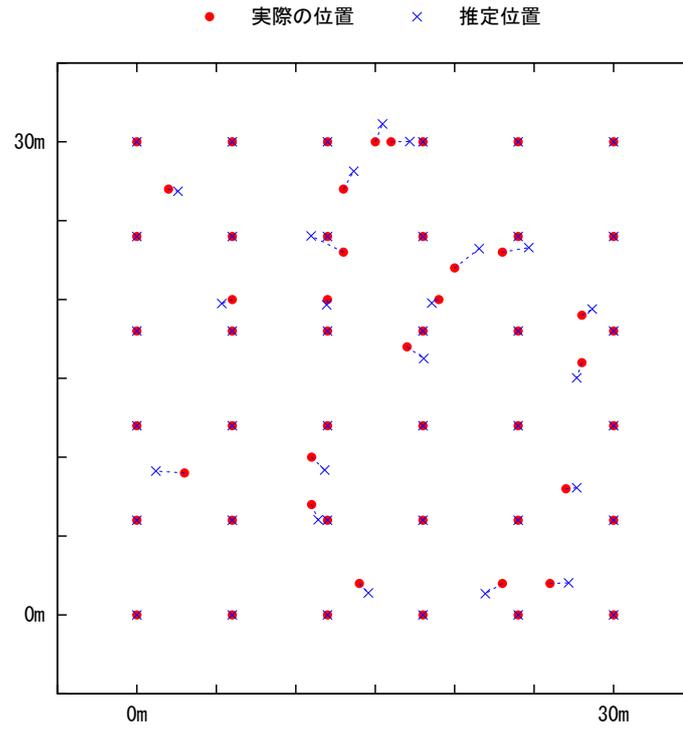


(c) 位置参照点 16 個

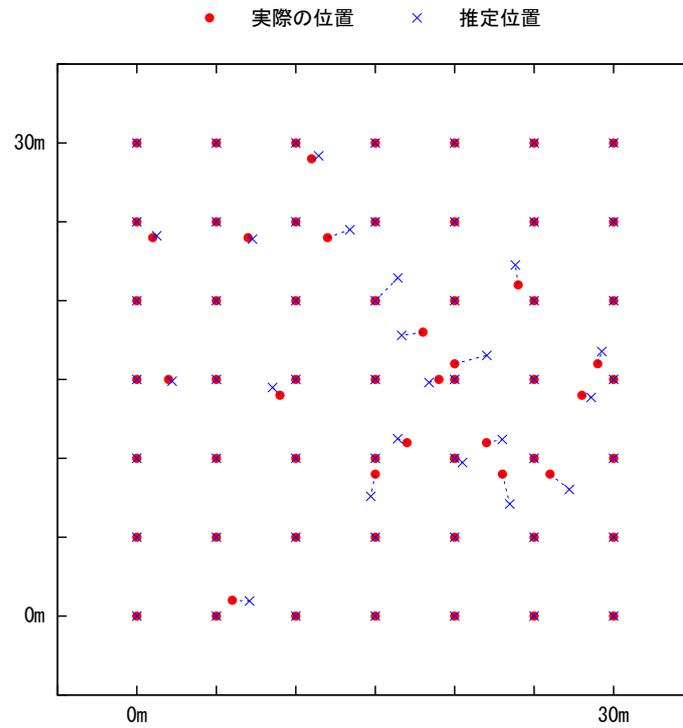


(d) 位置参照点 25 個

図 5.4: 参照点数に関した実際の位置と推定位置の比較 (続き)



(e) 位置参照点 36 個



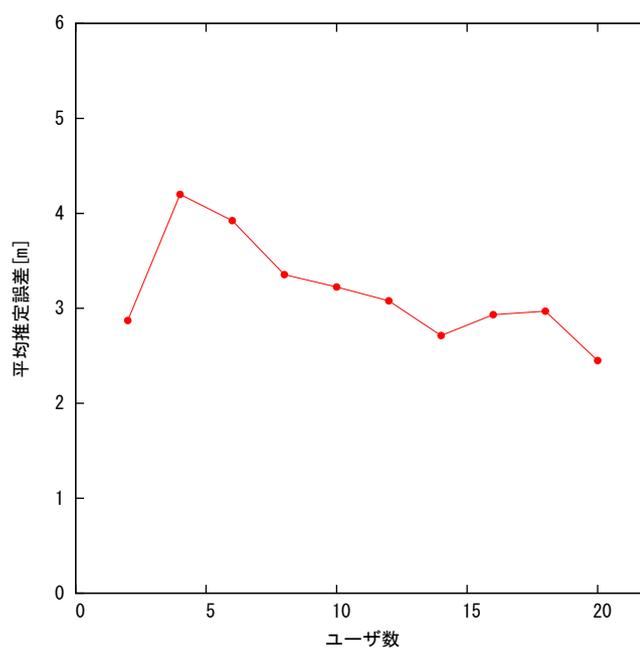
(f) 位置参照点 49 個

図 5.5: 参照点数に関係した実際の位置と推定位置の比較 (続き)

図 5.3~5.5 より、位置参照点数が増えると、ユーザの推定の精度が向上しているのが分かる。

5.4 推定ユーザ数依存性

5.1 と同様の条件のもと、推定ユーザ数を 2 から 20 までと 20 から 100 まで変化させたときの平均推定誤差を算出しグラフに示す。図 5.6 では、推定ユーザ数の増加に伴い、平均推定誤差が小さくなることが確認できる。これは、ユーザ数が増え、Stress Majorization に与える接続情報が多くなることで、より適切な相対座標グラフが描画されることに起因する。推定ユーザ数が 2 のとき平均推定誤差が他の推定ユーザ数（例えばユーザ数 4 人）のときよりも小さい。これは、推定ユーザ数が 2 のとき、推定ユーザの配置によっては、相対座標グラフが座標変換により適したパターンになり、その誤差が 1m 弱になるためだと考えられる。

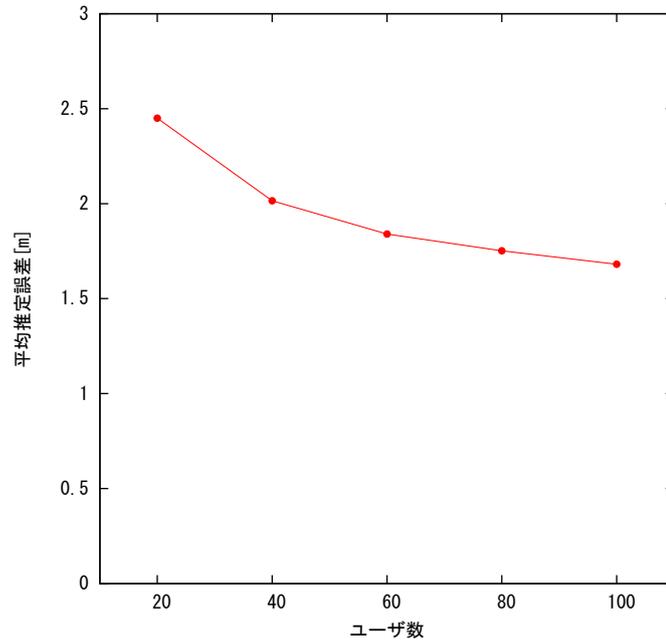


(a) ユーザ数 2 から 20

図 5.6: 接続距離と推定ユーザ数の関係

推定ユーザ数が 20 人以上に増えた場合でも、図 5.7 に示す通り、平均推定誤差がわずかに小さくなる。しかし、実用的な問題で考えると、30m 四方の部屋に推定ユーザが 20 人以上存在することは稀であるので、提案手法の精度はアンカーノード 5 個設置の場合であると

推定ユーザ数に関わらず，誤差 2m から 4m 程度で運用可能と言える．



(b) ユーザ数 20 から 100

図 5.7: 接続距離と推定ユーザ数の関係 (続き)

5.5 接続距離依存性

ユーザ同士が接続されていると判断する距離（ビーコンの届く距離）の変化が推測精度に影響を与えるのかを調べた．図 5.8 に 5.1 と同様の条件のもと接続距離を 10m から 30m まで変化させたときの，平均推測誤差を示す．

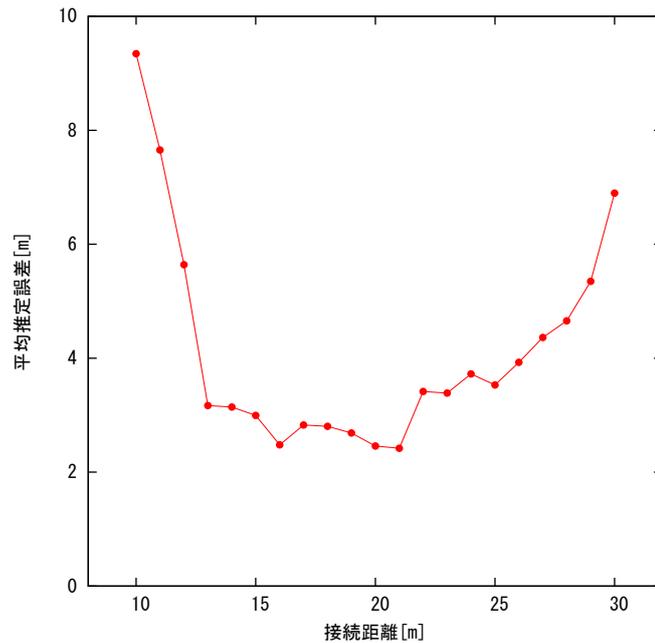


図 5.8: 接続距離と誤差の関係

接続距離が10m から13mの間では接続距離の増加に伴い、平均推定誤差が減少している。しかし、14m から21mの間では誤差に変化があまり見られない。また、22m から誤差が増加している。この事実から、接続距離の大きさと誤差は必ずしも比例関係ではないことが言える。ある接続距離までは近接情報の情報量が増えるため、誤差が減少するが、接続距離を大きくしすぎると、本来遠い位置に存在するユーザ同士が近い位置にいるという誤った情報が与えられてしまい、適切な相対位置マップが描画できないため誤差が増加するのだと考えられる。そのため、精度の高い推定をするには、適切な接続距離を設定する必要がある。また、屋内の広さに関して最適な接続距離が変わると予想できる。接続距離が極端に小さい場合、どのノードとも「接続」しないノードが出てくるため、適切な相対座標グラフが描画できず、位置推定の精度が下がる。

5.6 シミュレーション条件2

条件を変更し、位置参照点依存性と推定ユーザ数依存性、接続距離依存性について同様の考察が得られるか検証した。条件は以下とした。

- 100m 四方の部屋に 30 人の位置推定ユーザが存在
 - ユーザは部屋内にランダムに位置する
- 位置参照点を部屋内に 5 個設置
 - 部屋の四隅にそれぞれ 1 個設置
 - 部屋の中央に 1 個設置
- 距離 50m 以内のユーザ同士は「接続している」

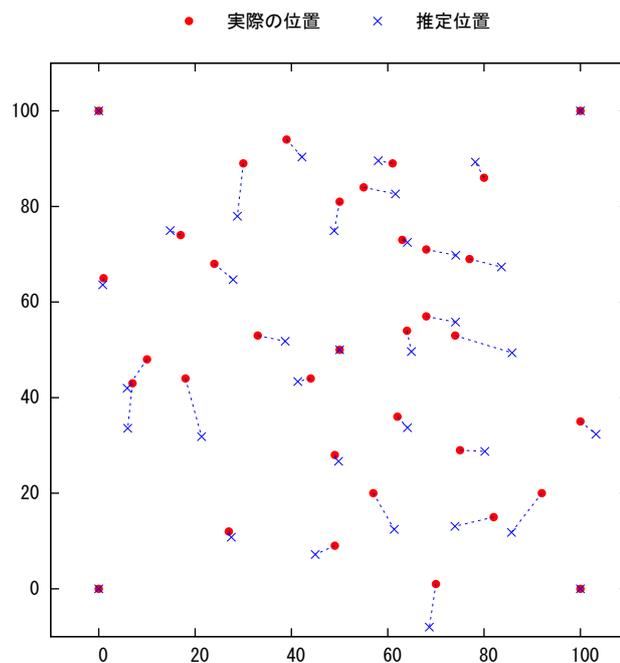


図 5.9: 実際の位置と推定位置の比較

以上の条件でも良好な位置推定が行われていることが確認できる。

次に、5.3 と同様、位置参照点を 4,9,16,25,36,49 個配置したときの平均推測誤差を示す。

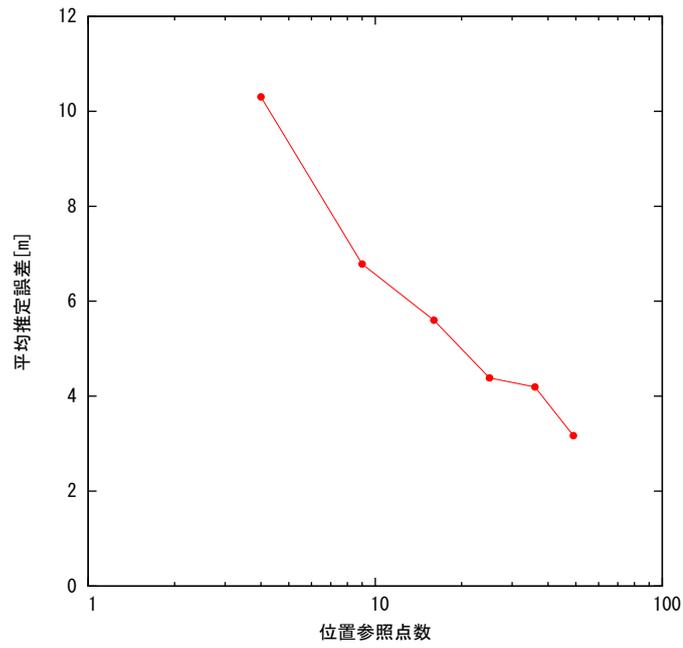


図 5.10: 位置参照点数と誤差の関係 (条件 2)

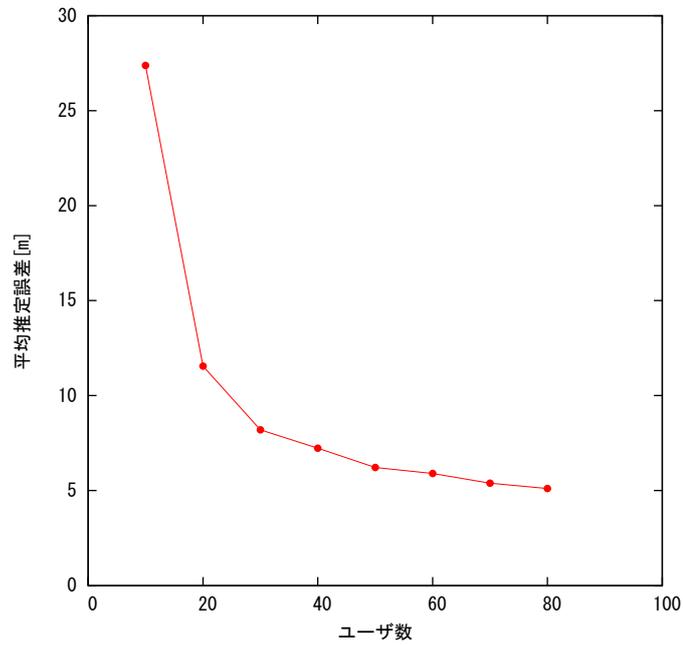


図 5.11: 推定ユーザ数と誤差の関係 (条件 2)

図 5.11 より, 推定ユーザ数を増やすと推定誤差が減少することが確認できた.
次に, 接続距離を 20m から 100m まで変化させたときの平均推定誤差を示す.

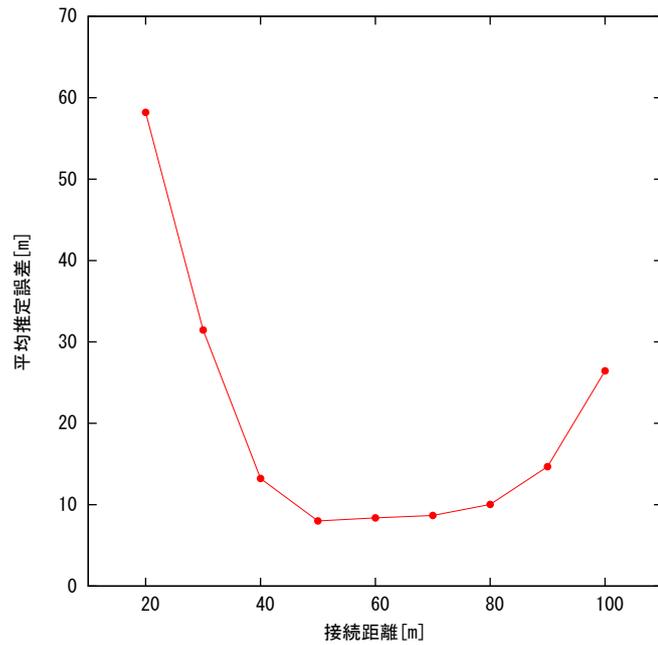


図 5.12: 接続距離と誤差の関係 (条件 2)

図 5.12 より、提案手法の推定誤差が位置参照点数の 2 乗に比例して減少していることが分かる。これは 5.3 と同じであるため、部屋の大きさと、位置参照点数に関する推定誤差の減少に、従属性がないと言える。図 5.12 を見ると、接続距離が 20m から 50m の間では接続距離の増加に伴い、平均推定誤差が減少している。しかし、50m から 70m の間では誤差に変化があまり見られない。また、80m から誤差が増加している。図 5.8 と比較すると、似た U 字型のグラフとなっていることから本条件においても同じ考察が言える。また、部屋の大きさと、接続距離と誤差の関係に従属性がないことが言える。100m 四方の部屋では、図 5.12 より、適切な接続距離は 50m から 70m と言える。30m 四方の部屋では適切な接続距離は 14m から 21m であった。この事実から、一辺長さの 50% から 70% の距離が適切な接続距離であると考察できる。例えば 10m 四方の部屋を提案手法で位置推定する場合、その一辺の長さの 50~70% なので、接続距離を 5~7m にすると精度の良い推定が行えると予想できる。

第6章 結論

シミュレータで位置推定の誤差を従来手法と比較した結果，提案手法の誤差は2.45m，セントロイドアルゴリズムでは4.12mであった．提案手法では，位置推定の誤差を従来手法の半分程度に抑えることができた．従って，提案手法により位置推定精度の向上が可能であると言える．近接情報を利用する本手法は，3章で紹介した Zigbee，または Bluetooth 規格の無線端末での実用化が現実的である．提案手法は位置参照点を多く設置することで，誤差が小さくなり，位置推定の精度が上がることが分かった．また，推定ユーザ数を増やすことでも位置推定の精度が上がるということがわかった．これらは，多くの接続情報を取得できるためであるから，提案手法の精度を向上させるためには，接続情報を多く取得する必要があると考察できる．提案手法は位置推定を行う環境によって，適切な接続距離が変わるため，利用する環境に合わせた接続距離の設定をする必要があるが，本研究によって，測定環境が四方の部屋である場合は，部屋の一边の50~70%の長さが適切な接続距離であることが判明した．測定環境や，ユーザ数に関して適した接続距離があると予想されるため，今後はこの予想について調査していく．

謝辞

本研究を進めるにあたり，ご指導頂いた卒業論文指導教員の塩田茂雄教授に深く感謝します。また，竹之内先輩には多くの助言を頂きました。ありがとうございます。最後に，お世話になった塩田研究室の皆様感謝致します。

参考文献

- [1] 大槻知明, “位置推定技術,” 信学技報, pp.3, 2006
- [2] 辻宏之, “アレーアンテナを用いた屋内外の無線局位置推定の実験的検証,” 電子情報通信学会論文誌 B, Vol.J90-B, No. 9, pp.784-796, 2007
- [3] N. Bulusu, J.S. Heidemann, D. Estrin, and T. Tran, “Selfconfiguring localization systems: Design and experimental evaluation,” *ACM Trans. Embedded Comput. Syst.* vol. 3, no. 1, pp. 24-60, 2004.
- [4] D. Niculescu and B. Nath, “DV based positioning in ad hoc networks,” *Telecommunication Systems*, vol. 22, no. 1-4, pp. 267-280, 2003.
- [5] T. He, C. Huang, B.M. Blum, J.A. Stankovic, and T.F. Abdelzaher, “Range-free localization schemes in large scale sensor networks,” *Proc. of ACM/IEEE MOBICOM 03*, 2003.
- [6] 島村和希, “センサの相対位置推定のための最適化手法,” 千葉大学院工学研究科修士論文, pp.14, 2014
- [7] 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,” *International Conference*, pp.3, 2006
- [8] 中尾浩一, “屋内測位技術の動向について,” http://www.apptec.co.jp/technical_report/pdf/vol22/treport_vol_22-09.pdf,(参照 2017/1/27)
- [9] 国立研究開発法人情報通信研究機構, “UWB を利用した高精度の屋内測位システムを開発,” <https://www.nict.go.jp/press/2014/05/26-1.html>,(参照 2017/1/20)
- [10] 佐藤智美, 小宮山哲, 下田雅彦, 劉渤江, 横田一正, “Bluetooth の電波強度を用いた位置推定方式の検討,” DEIM Forum 2011 B9-4

- [11] E. Gansner, et al., “Graph drawing by stress majorization,” in *Graph Drawing*, ser. Lecture Notes in Computer Science. Springer, 2005, pp.239-250.
- [12] B. Horn, et al., “Closed-form solution of absolute orientation using orthonormal matrices,” *Journal of the Optical Society of America*, vol.5, no.7, pp.1127-1135, 1988.