

平成 23 年度 特別研究報告書

端末間通信による 屋内位置推定

龍谷大学 理工学部 情報メディア学科

学籍番号 T080387 井上亮弥

指導教員 三好 力 教授

内容梗概

近年、さまざまな機能を搭載したスマートフォンの普及が進んでいる。本論文ではその機能の中の GPS 機能に着目した。現在では GPS を使い誰もが容易に現在位置を知ることが可能となったが、それは屋外での話であり、屋内では測定することが困難であり、時にはできないこともある。既存技術で屋内位置検索システムはあるが、別途基地局や専用端末を要しさらに設置費用は最小規模でもかなり高額である。そこで本実験では基地局を用いることなく、スマートフォンなどの端末間通信のみで 2 次元座標の推定、および 3 次元座標の推定を試みた。

実験結果から 2 次元座標の推定は 2 つ以上のノードと通信を行うことができればほぼ正確な座標を推定できることがわかった。3 次元座標の推定は仮定した通信距離と実際の通信距離との誤差が $\pm 4\%$ 以内の範囲のとき、推定した高さはおよそ $\pm 10\%$ 以内の誤差の範囲であることがわかった。しかし、通信距離の誤差がこれより大きくなると高さの推定値は分散してしまう。

目次

第一章 はじめに.....	1
第二章 既存技術.....	2
2.1 GPS.....	2
2.2 GOOGLE MAP.....	3
2.3 MANET.....	3
2.3.1 マルチホップ通信.....	4
2.3.2 フラットティング.....	4
2.4 屋内位置検索.....	5
2.4.1 日立 AirLocation.....	5
2.4.2 システム構成.....	6
2.4.3 測定方法.....	6
2.5 プレース・ステッカー.....	8
2.5.1 概要.....	8
2.5.2 仕様.....	8
第三章 提案手法.....	9
3.1 既存技術における問題点.....	9
3.2 提案手法.....	9
3.2.1 2次元測位.....	10
3.2.2 3次元測位.....	11
第四章 実験.....	13
4.1 実験概要.....	13
4.2 実験結果.....	16
4.2.1 2次元測位.....	16
4.1.2 3次元測位.....	17
第五章 おわりに.....	18
謝辞.....	19

参考文献..... 20

付録..... 21

第一章 はじめに

近年、スマートフォンの普及が進んでおり、今や携帯電話は電話やメールなどの連絡手段だけにとどまらず、Web ブラウジング機能の向上、携帯型ミュージックプレイヤーとの一体化、GPS (Global Positioning System) による現在位置の検索、ナビゲーション機能の向上、および多種多様なアプリケーションによる機能の拡張など様々な機能を搭載されているのが、もはや当たり前のようにになっている。その中でも、本研究では GPS による現在位置の検索に着目する。現在、GPS により現在地の推定は容易にできるようになった。GPS の精度は使用する端末、環境によって異なるが、数メートルから数十メートルぐらいである。しかし、それは屋外における話であり、上空を見ることができない屋内や、高層ビルが密集する都心となると測定は困難であり、場合によってはできないこともある。しかし、東京、大阪などの都会だけでなく、各地方でも建物のビル化や大型ショッピングモールの開設など便利にはなるものの建物の構造は複雑になり、施設や店舗内に備え付けられた地図だけでは不十分であることも多い。そこで、用いられているのが屋内における現在位置の推定システムである。この設備があることにより、広い建物内の GPS が届かないような場所でも自分が建物のどこにいるのかといった情報を取得することができ、施設の地図情報と現在組み合わせることで、目的の場所やお店に迷うことなく行くことができるようになり、また今いる店舗の電子チラシを自動で表示させるなどといったことも可能になってくる。つまり、消費者だけでなく、各店舗の経営者にとっても有益といえる。現在、位置検索を可能とする技術はいくつか存在するが、高性能な基地局の設置を必要とするため、コスト面で莫大な費用がかかり、設置にも手間がかかる。そこで、本論文では基地局を用いず、移動型携帯端末のみを用いて屋内現在位置検索をおこなう方法を提案する。

第一章では研究背景,研究の目的について述べる。

第二章では本論文で用いる通信をはじめ、既存する技術を述べる。

第三章では、2章で紹介した既存技術の問題点を明確化し、そこから見る改善のための手法について述べる。

第四章では、本研究で行った実験の目的と概要を述べ考察を述べる。

第五章では、本実験での結論、今後の課題を述べる。

以上を本論文の構成とする。

第二章 既存技術

本章では本研究で用いる端末間通信の説明、及び既存の技術の解説を行い、問題点を挙げる。

2.1 GPS

GPSとは、Global Positioning Systemの略で、全世界的な位置測定システムである。GPSは24時間いつでも、地球上のどこにおいても、誰もが簡単に使用でき、しかも高精度に位置を測定することができるシステムのことを示し、上空約20000mのところを飛行している24個以上の専用の人工衛星と地上施設で構成されている。また、12個の飛行軌道が設定されており、それぞれの人工衛星が約12時間周期で飛行しており、このことにより24時間いつでも現在位置の測定をすることができる。測定には最低でも4つの衛星からの電波が届く必要があり、内3基で3次元的位置(X、Y、Z)より3次元測位を行い、残りの1基で正確な時間(T)を取得しこれら4つの値から4元連立方程式により値を求めることにより、位置を割り出している。

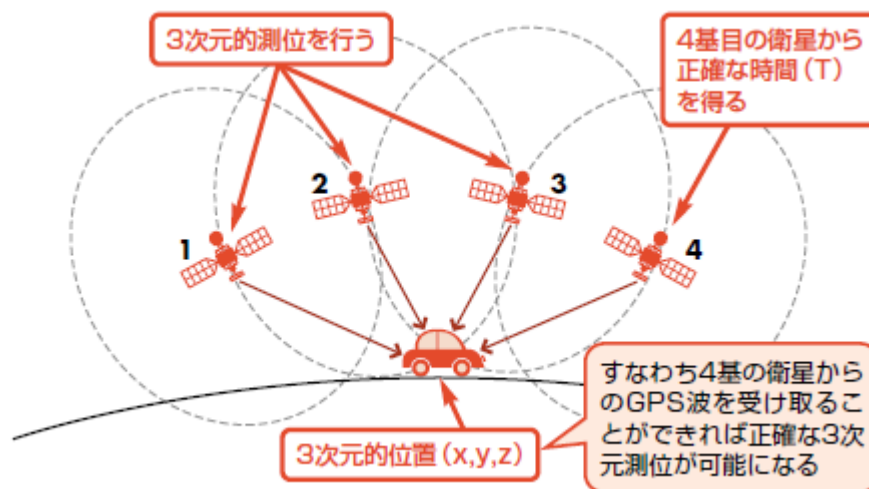


図1 GPSによる測定方法[引用1]

2.2 Google Map

Google Map とは Google 社が提供している全世界地図サービスであり、多くのスマートフォンに初期アプリとして搭載されている。このアプリケーションの大半には現在地推定機能が備わっているが、これは GPS だけでなく、主に飲食店や携帯販売店などに設けられてフリーの Wi-Fi スポットの設置されている位置をデータベース化し、どの Wi-Fi の通信範囲内に存在するのか、また Cell ID (携帯電話の基地局) での基地局の範囲内にいるのかという 2 つから現在地を割り出す際の精度を上げている。

2.3 MANET

MANET とは、従来のインターネットとは異なり、イーサネットや無線 LAN のアクセスポイントといった基地局などのインフラを必要とせず、自律的に形成できるネットワークのことを指し示す。つまり、端末同士が直接接続して構成するネットワークのことである。事前にインフラを用意せずとも、その場でネットワークを構築することが可能である。また、無線通信を使ってデータを端末から端末へパケット・リレーのように手渡すことから、「マルチホップ・ネットワーク」とも呼ぶ。無線 LAN の普及に伴って注目を集めている。今日、MANET の技術はニンテンドーDS や PlayStation Portable などの携帯型ゲーム機や近年爆発的に普及しているスマートフォンなどに用いられている。

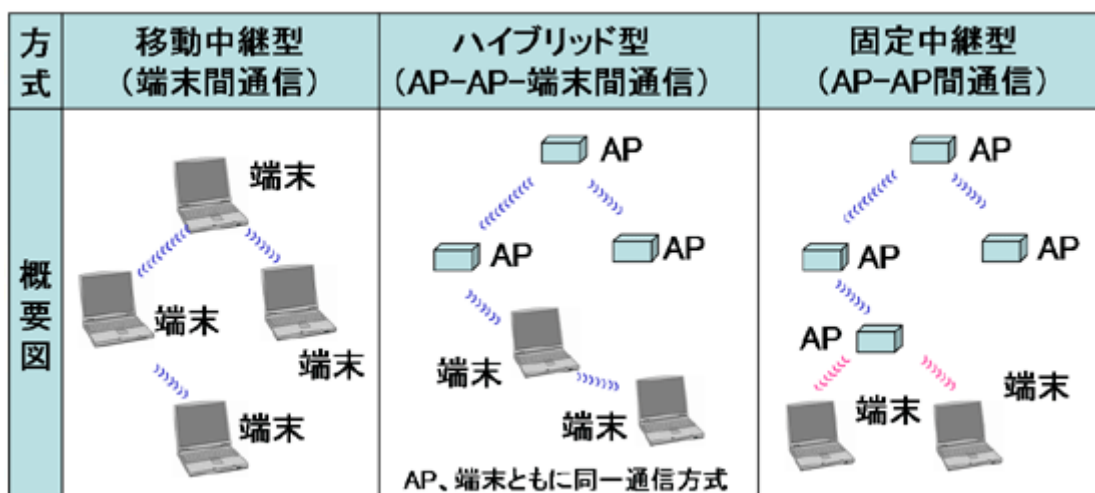


図2 ワイヤレスアドホックネットワーク運用形態[引用 2]

ワイヤレスアドホックネットワークは自律分散型であり、移動中継型、ハイブリッド型、固定中継型の 3 種類のネットワークの運用形態がある。(図 1)

NTT では少ない工事稼働でネットワーク構築が可能で、かつ安定なサービス提供が可能なアクセスポイント (AP) 間をワイヤレスアドホックネットワークで構築する固定中継型あるいはハイブリッド型での運用を考案しており、これらの高ス

ループット化、VoIP・映像系アプリに対応するための QoS 制御、ハンドオーバー技術、またセキュリティ確保などの研究開発がおこなわれている。

(1) 高スループット化

単一 Ch でのワイヤレスアドホックネットワーク構築に無線 LAN (IEEE802.11) を使用する場合、隠れ端末問題、さらし端末問題によってスループットが低下する。このため、複数 Ch 運用が有効で、自律的な Ch 割当、ルーティング、効率的なパケット転送方法が課題となる。

(2) QoS 制御

VoIP や画像情報送信などの利用が考えられ、ワイヤレスアドホックネットワークにおいても、優先制御・帯域制御技術などの QoS 制御が課題となる。

(3) ハンドオーバー

VoIP や画像情報送信などの利用シーンとしても歩行程度の速度での移動を考慮した場合に、アクセスポイントをまたがって移動することも考えられるので、可能な限り品質劣化を抑えるためのハンドオーバー技術が必要となる。

(4) セキュリティ確保

ワイヤレスアドホックネットワークでは盗聴防止、端末のなり済まし防止、パケット改竄防止などのセキュリティ確保が必須であり、アドホックネットワークでは、さらに不正なアクセスポイントの設置が容易であること、また認証パケットなどを模擬した不正パケットの大量侵入により無線帯域が大幅に消費され通信が妨害される可能性がある。このため、アクセスポイント自身の認証、不正パケットの検出・転送防止策の重要となる。

2.3.1 マルチホップ通信

マルチホップ通信とは送信元端末から通信先端末まで電波が届かないなどの状態の場合において、一台あるいは複数台の端末を中継し、データを伝送することである。ほかにも、アドホックネットワークでは、端末が移動することも前提として開発されている。一度経路を確立してもその端末が移動してしまうことがあり、もう一度、同じ経路が利用できるとは限らない。そこで 2.3.2 で示す手法がある。

2.3.2 フラッティング

アドホックネットワークではデータを転送する際に、端末がどの端末と通信できるネットワークが構築されているかを知る必要がある。各端末が通信経路を作成するために行う技術がフラッティングである。これはデータを送るためのものではなく、経路発見のための経路制御パケットを通信するもので、ブロードキャストで次から次へと転送し経路を発見していくためのものである。送信ノード (S) から宛先ノード (G) にデータを送るための通信経路の探索を行う。そのときに制御をおこなうプロトコルを 2.4.1、2.4.2、2.4.3 に示す。

2.3.2.1 リアクティブ型プロトコル

リアクティブ型プロトコルでは、通信要求がなされた場合に限り、経路探索を行う。よって、通信要求がない場合は経路制御パケットは一切送信されていない。

しかし、短時間のうちに複数の通信要求があると大量のデータパケットと経路制御パケットが通信されることとなり、ネットワークに負荷をかけることとなる。

そのため、経路探索によって確立された通信経路を再利用することによりネットワークの負荷を軽減させている。この型は各端末の移動が頻繁に行われるネットワークの場合に有効である。

2.3.2.2 プロアクティブ型プロトコル

プロアクティブ型は事前に通信経路を確立し、定期的に各端末において経路情報の交換を行う。事前に通信経路を確立しているために通信要求があればすぐにデータのやり取りを行うことができる。この型は、各端末の移動が少ないネットワークに対して有効である。

2.3.2.3 ハイブリッド型プロトコル

リアクティブ型、プロアクティブ型の両方の特性を持つものがハイブリッド型である。中継回数が少ない端末に関してはプロアクティブ型で通信経路の構築をおこない、中継回数が多い端末に対しては、リアクティブ型で通信するなどを行う型である。

2.4 屋内位置検索

2.4.1 日立 AirLocation

日立 AirLocation は無線 LAN、IEEE802.11.b 準拠の位置情報検索システムである。人やものの位置を可視化するとともに、無線 LAN ならではの多様なアプリケーションがある。特徴としては、

- ・ダイバーシティ方式による 1～3 m の高精度位置検知
- ・IEEE802.11b による、凡庸無線 LAN 端末そのまま使用
→これにより設置後もその端末を用いての通信が可能になる。
- ・3 種類の測位方式によりより効率的なシステム構築が可能
- ・位置をリアルタイムにビジュアル化をおこなう

2.4.2 システム構成

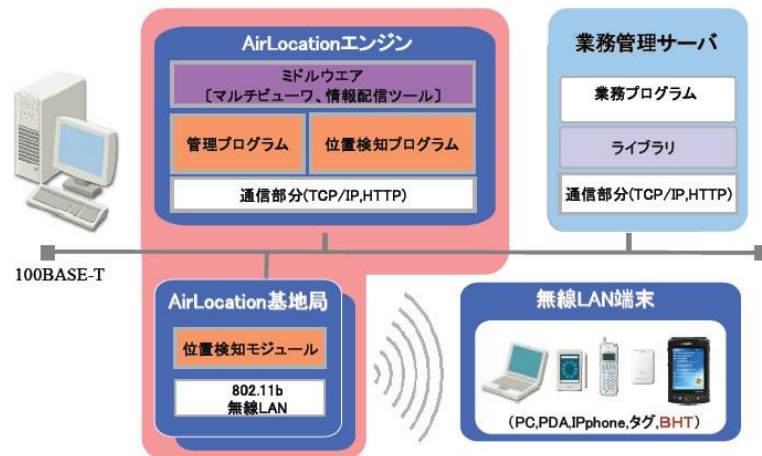


図3 システムの構成[引用 3]

上記図3のように基地局から端末に通信することにより位置情報を取得する。

2.4.3 測定方法

・2次元測位方式

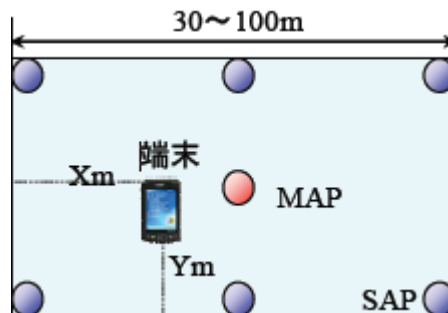


図4 2次元測位方式

- ・ 端末の電波をすべてSAPで受信
- ・ 三辺測量でX,Y座標を測位
- ・ 基地局は5台以上(推奨7台以上)
- ・ 使用例は倉庫等の広域エリア

・1次元測位方式



図4 1次元測位方式

- ・ 端末電波を SAP で受信
- ・ 一辺測量で線上座標を測位
- ・ 基地局は 3 台以上
- ・ 使用例は回廊等の線エリア

・ アソシエイション方式

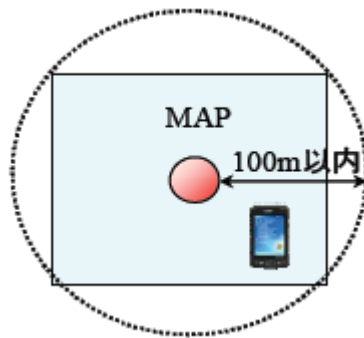


図 6 アソシエイション方式

- ・ 端末情報を MAP で受信
- ・ 端末接続情報を利用
- ・ 基地局は一台
- ・ 使用例は会議室などの小部屋エリア

2.5 プレース・ステッカー

2.5.1 概要

これは立命館大学とローム、電通国際情報サービスが行動開発をおこなったもので、従来難しかった地下街やビル内部で高精度の測位し、屋内でスマートフォン(多機能携帯電話)の位置情報サービスを利用できるようにする測位装置である。

2.5.2 仕様

高速無線通信「W i F i (ワイファイ)」を用いており、サイズは縦 5 センチ、幅 6.5 センチで、スマートフォンと交信して位置を特定する。通常のW i F i 通信は出力が高く、床や壁の反射で高精度測位ができないため、出力を 10 分の 1 に抑えている。また、室内照明で発電するロームの「色素増感型太陽電池」を組み合わせ、外部電源なしで独立稼働する。



図7 プレース・ステッカー[引用 4]

第三章 提案手法

3.1 既存技術における問題点

前章でも記したが、GPS による現在位置の推定は屋外に限るものであり、屋内ではその効果はあまり発揮されない。しかし、近代化が進み建物の構造が複雑化し、また規模自体が大きくなってきている。そこで、必要となってくるのが屋内現在位置測定である。すでにそれに関する技術はいくつか開発されている。しかし、そのどれもが基地局の設置を必要とし、そのことに莫大な費用と、時間を投資する必要がある。例をあげると 2次元測位システムにおける最小規模の構築でも「500万円」という高額な設備投資費用が必要となり、また中には専用の携帯型移動端末が必要なシステムも存在する。そこで、本論文では、基地局の設置にかかる費用と時間、及び汎用性に着目し、問題点とする。

3.2 提案手法

3.1 の問題点より基地局を用いず、汎用性の高いスマートフォンなどの移動型携帯端末（以下ノードと記述）のみを用いて屋内における 3次元の現在地測位を行う手法を提案する。

この提案手法では屋内に存在する推定するノード（以下推定ノードと呼ぶ）の座標を GPS による位置情報を取得できる屋外を移動しているノード（以下発信ノードと呼ぶ）との間で端末間通信を行い、GPS の情報を提供してもらうことにより、屋内位置を推定するというものである。まず、はじめに 2次元の平面座標を推定し、その推定結果を用いて、高さを含める 3次元座標を推定する。

3.2.1 2次元測位

図8は提案手法の2次元位置推定を示した概念図である。×点は推定ノードの本来の位置を示している。黒点は発信ノードの初期位置を示し、矢印はその進行方向及び進行した方向を、赤い範囲は推定ノードと端末間通信が可能な範囲を示している。推定ノードは発信ノードが通信可能範囲内、つまり、図の赤い範囲に差し掛かった時、また範囲外に離れる時の2点（図8中の赤点）のGPSによる位置情報を取得する。これにより取得した2点間の中点を計算で割り出し、そこに図中に青線で示した垂線を計算する。これと同様に進行角度が異なる発信ノードが通信可能範囲内に差し掛かった時、及び通信可能範囲から外れるときのGPSによる位置情報を発信ノードから取得し同じく垂線を求める。これら求めた2垂線の交点を推定ノードの推定位置とする。

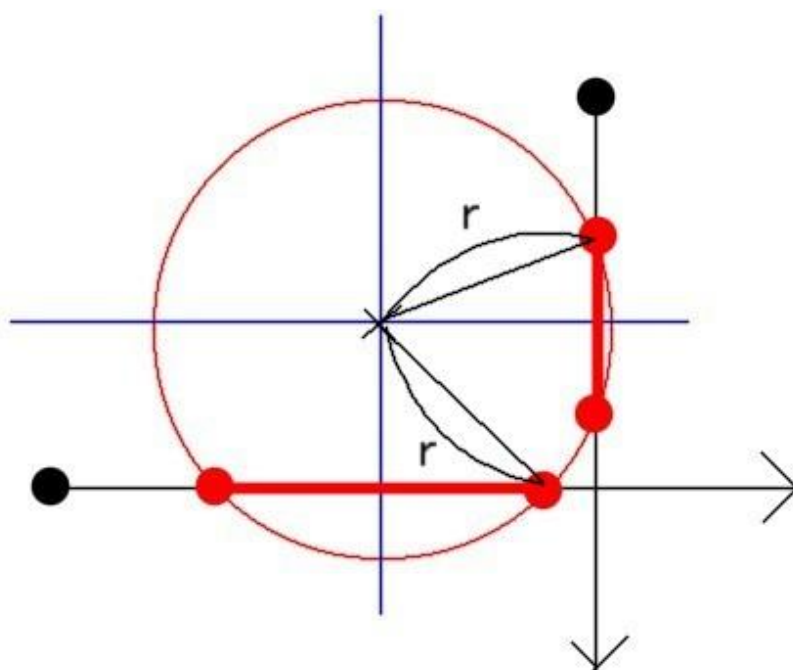


図8 提案手法による2次元測位

3.2.2 3次元測位

本論文では2次元の位置推定に加え、3次元の位置推定を行う方法を提案する。図9は3次元位置推定の概念を示した図である。赤点は推定ノードの位置を示し、黒枠で示した建物の内、地表からの高さ h のところに存在する。また青点、水色点はそれぞれ発信ノードが推定ノードと端末間通信を開始した点と通信が途切れる直前の点を示したものであり、その2点の midpoint を黄点で示す。この時、それぞれの発信ノードは建物の壁際に沿って移動しているとする。また、紫点線は通信可能距離 r を示しており、推定ノードと各発信ノードは通信距離 r で通信を行うとする。

推定ノードの平面座標は提案手法の2次元位置推定を用いて推定できているとする。

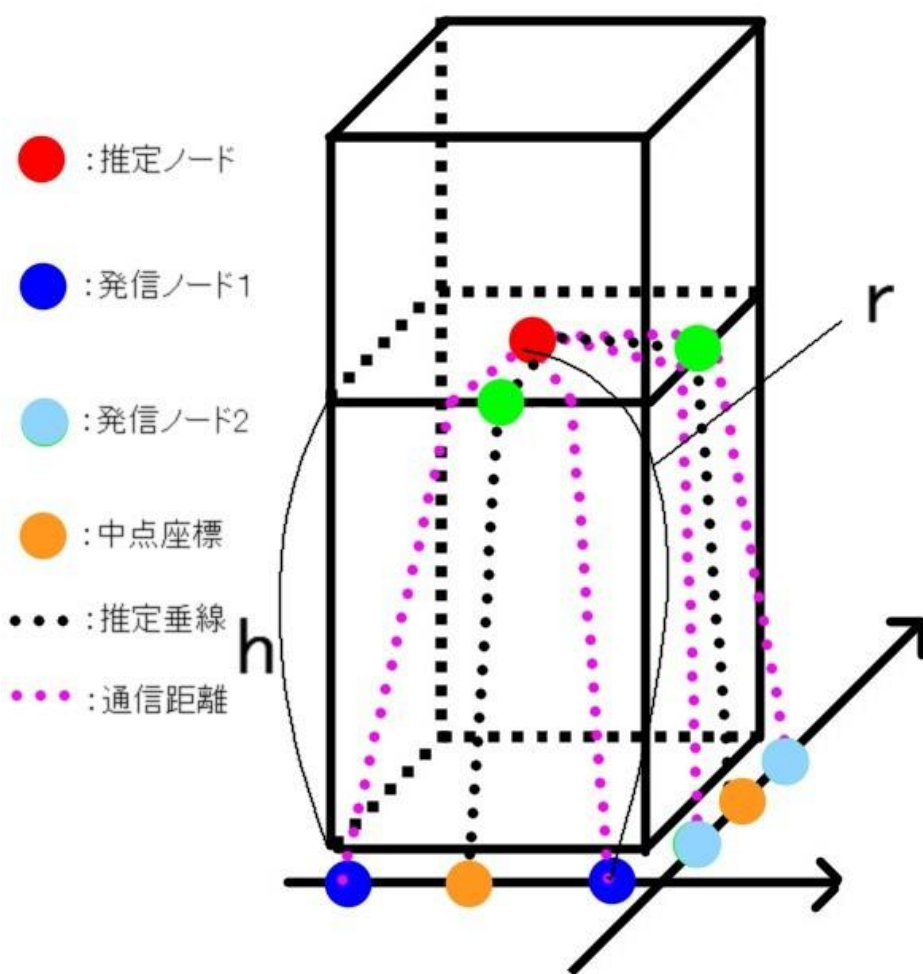


図9 3次元測位法

図 10 は図 9 の紫点線で示した赤点ノード青点ノード間における通信を平面化したものである。

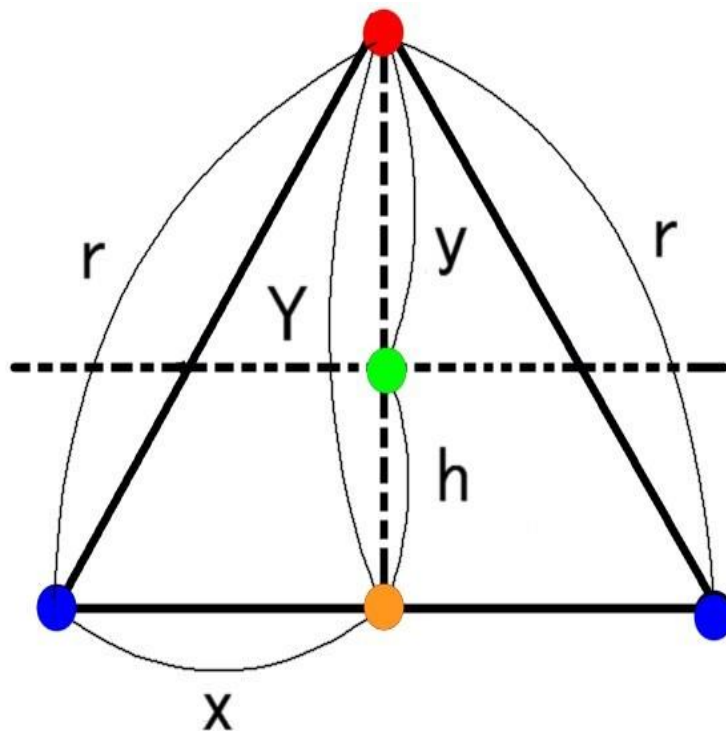


図 10 赤点ノード青点ノード間通信

図 10 において赤点が推定ノード、青点が通信開始、及び通信が途切れた座標を示し、これらの座標は 2 次元位置推定の段階で分かっているとす。このとき、 r は一定で二等辺三角形になっているので

$$Y = \sqrt{r^2 - x^2} \dots \dots \dots (1)$$

また、窓際の発信ノードは窓に極めて近い位置に存在すると近似するので緑点と黄点は同一の 2 次元座標となるので 2 点間の距離である y は計算で求めることができる。

これらのことより高さ h は

$$h = \sqrt{r^2 - x^2} - y \dots \dots \dots (2)$$

で求めることができる。

第四章 実験

4.1 実験概要

提案手法では通信可能距離を r と固定して計算を行うが、実際はそれぞれの端末によって通信可能距離は異なる。そこで発信ノードから電波の出力電力量を取得できるものとし、その取得した値から推定距離 R と仮定できるとする。しかし、端末の使用されている状況や周りの環境により実際の通信距離にある程度の誤差が生じる。そこで、今回の実験では、仮定した通信距離 R を用いて算出した推定値と誤差を含めた実際の通信距離を用いて算出した実際の値の間に生じる誤差を調べる評価実験を行う。提案手法の 2 次元位置推定においては実際の座標と推定した座標との誤差の評価を行い、3 次元推定においては通信距離の誤差が高さの推定位置と実際の高さの誤差にもたらす影響について、シミュレーションによる評価実験を行う。今回のシミュレーションで用いるパラメータは以下の通りである。

表 1 シミュレーションパラメータ

フィールドサイズ	62×62×0
建物のサイズ	20×20×25
道路幅	1
通信可能距離	25
仮定通信距離と実通信距離との誤差	±10%
推定ノード数	1
発信ノード数	4
試行回数	1000

また、表 1 を視覚化したものが図 11 であり、赤点は推定ノード、青点は発信ノードの初期位置、矢印は進行方向を示しており斜線部が建物の範囲を示している。

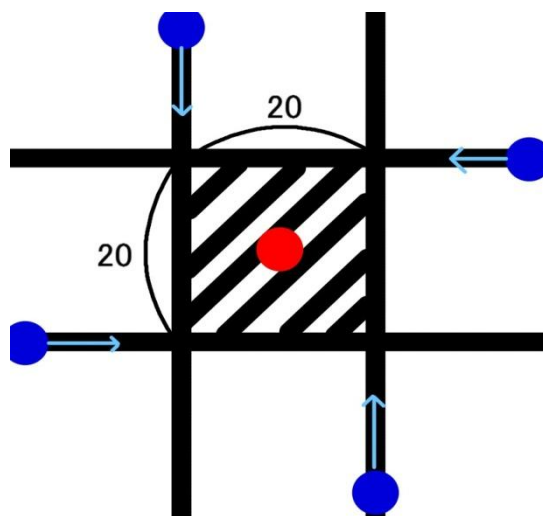


図 11 シミュレーション環境

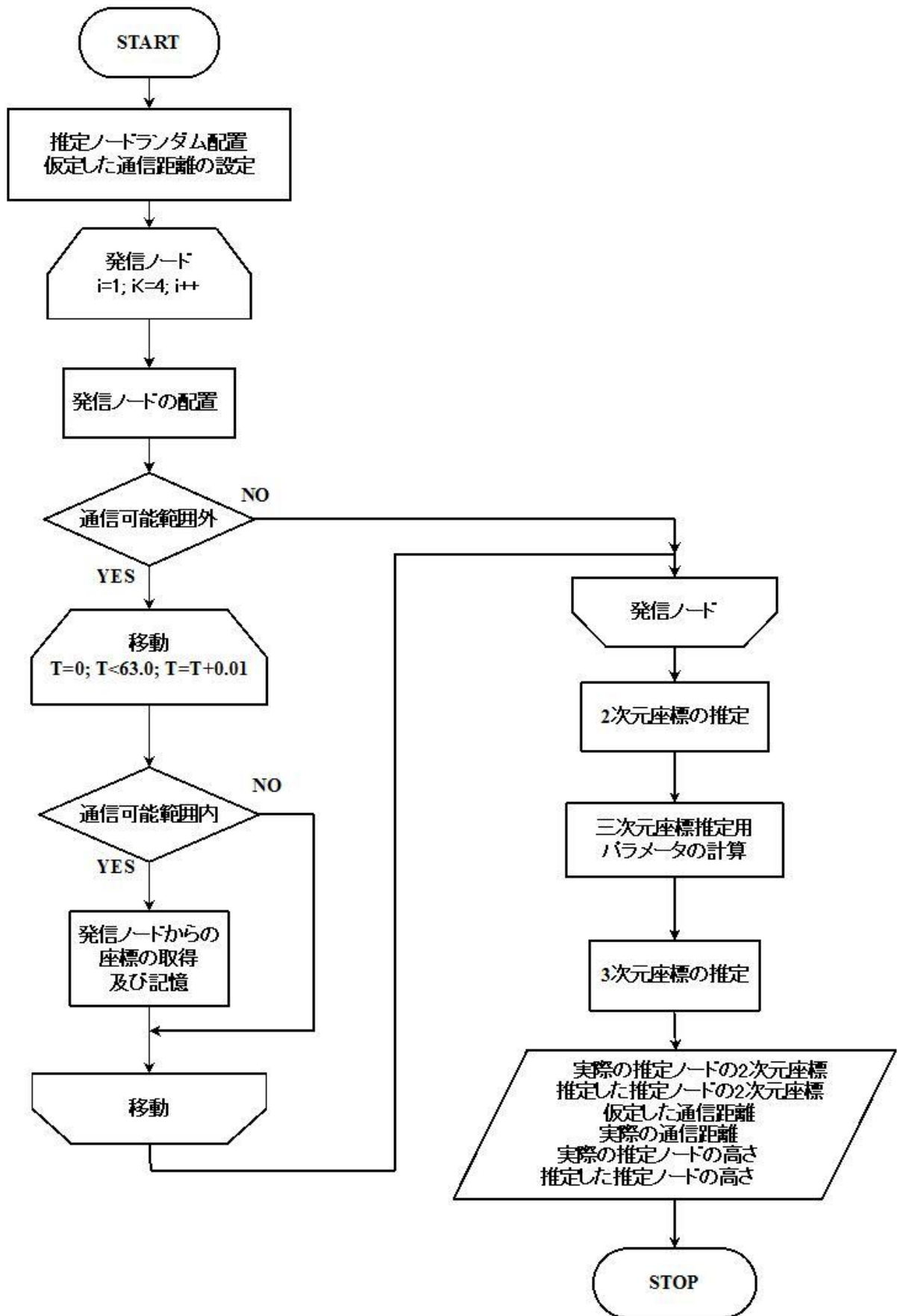


図 12 フローチャート

図 12 はプログラムの流れを表したものである。大まかな流れとしてはまず、ランダムで推定ノードを屋内に配置し、図 11 で示した場所に発信ノードを配置し、示した進行方向に移動させる。ノードを配置した際、もし推定ノードと発信ノードがすでに通信可能距離内に存在する時、本提案手法は通信可能距離内に入った際の GPS による位置情報を必要とするので、この場合はエラーとし、この発信ノードにおける一切の処理を中止し、次の発信ノードとの処理へと移行する。そうでなければ、0.01 刻みで移動を繰り返し、通信可能距離内に入ったとき、および通信可能距離外に出たときの GPS の位置情報を取得し、記憶しておく。発信ノードがフィールド端まで移動すると、次のノードに処理を移行し同様の処理を行う。配置した 4 つの発信ノードにおける処理が終了すると、記憶しておいた座標を元に 2 次元座標の推定を行う。この時、図 10 における x 、 y を算出し、それらの値と式(2)、式(3)を用いて実際の高さ h 、推定した高さ H を求める。このプログラムを 1000 回試行し、出力された実際の推定ノードの 2 次元座標、推定した推定ノードの 2 次元座標、実際の通信距離、仮定した通信距離、実際の高さ、推定した高さより評価実験を行う。

4.2 実験結果

4.2.1 2次元測位

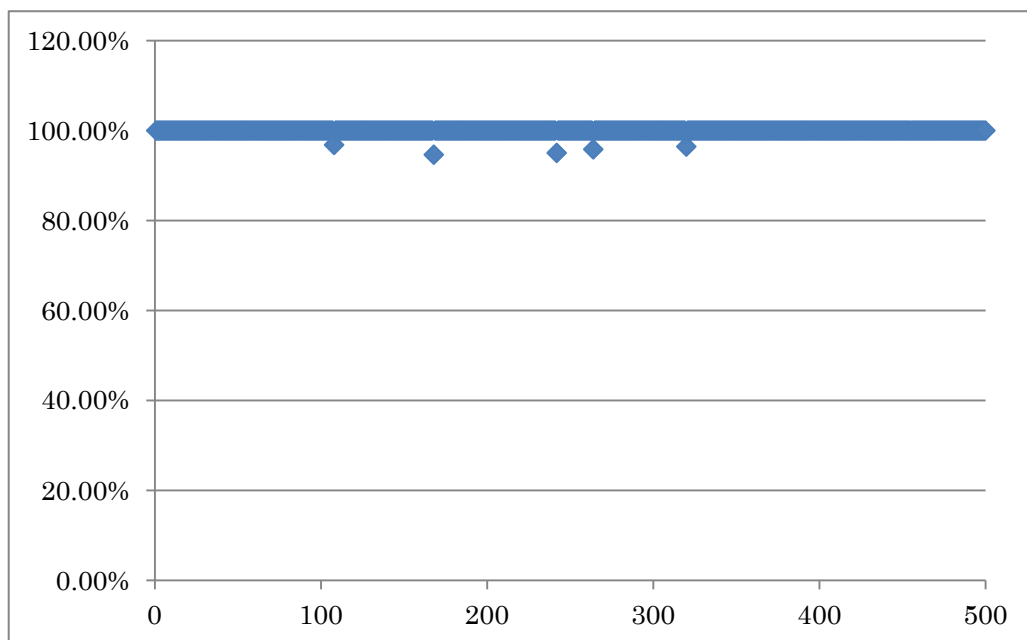


図 13 2次元推定における誤差比率

図 13は 4.1 節で示したプログラムを 500 回分の有効結果を得られるまで試行し、2次元推定における実際の座標と推定座標の誤差をグラフ化したものである。誤差の評価を表 2 にまとめる。

表 2 2次元推定評価実験における誤差

有効データ数	評価誤差	誤差範囲内データ数
500	5%	499
500	1%	495

これらの結果より 2次元推定で誤差はほぼ生じないことが分かった。

4.1.2 3次元測位

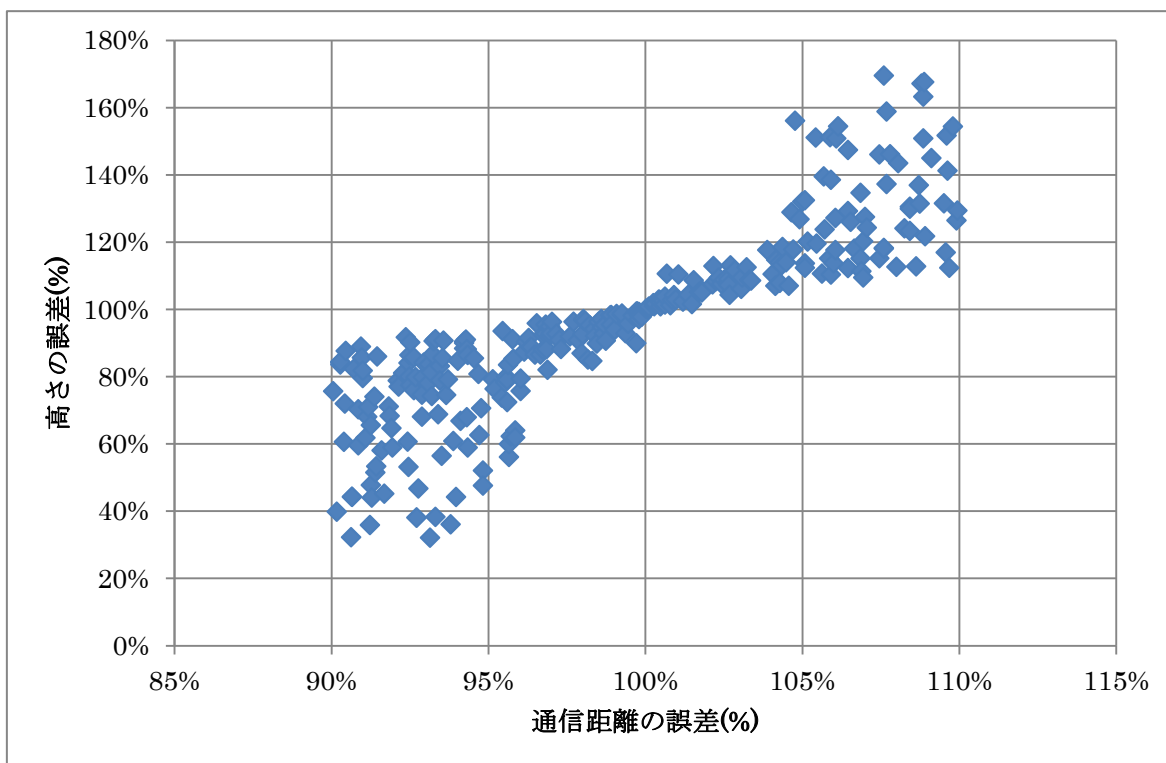


図 14 3次元推定における通信距離の誤差と高さの誤差の関係性

図 14 は 3 次元推定における仮定した通信距離と実際の通信距離の誤差の比率を横軸、及び推定した高さと実際の高さの誤差の比率を縦軸とし、関係性を示した図である。この図より通信距離の誤差がおよそ±4%以内の範囲で高さの誤差がおよそ±10%以内の範囲で比例することが分かった。また、それよりマイナス方向に誤差が増えると推定した高さは実際の高さよりずっと低い値に分散していることが見てとれる。また同様に、プラス方向に誤差が増えるとずっと高い値に分散することが図 4 から見てとれる。したがって、通信距離の誤差が±4%以上の誤差での推定値は正確さを大幅に欠くことがわかった。

第五章 おわりに

本論文では既存の屋内位置検索システムのコスト面や設置のための手間、また専用の携帯端末がいるなどの汎用性の低さなどの問題点と挙げ、基地局を用いることなく、スマートフォンなどの汎用性の高い携帯型移動端末同士の通信のみで屋内3次元位置推定を行う方法を提案した。

実験結果から2次元位置推定は通信状況さえ整えばほぼ正確に取得できることが分かった。3次元位置推定においては、通信距離の誤差が±4%以内の範囲のとき、高さはおよそ±10%以内の範囲で推定できることが分かった。

この方法だけではまだまだ誤差が大きいため、今後の課題として、端末に組み込まれている加速度センサーやその他のセンサーなどと組み合わせることで誤差範囲の減少化を図る。

謝辞

本研究を進めるにあたり、貴重な時間を割いてご指導いただいた三好先生をはじめ、議論を通して知識や示唆をいただいた同研究室の皆様、ならびに生活面で支えてくれた家族に深く感謝申し上げます。

参考文献

- [1] http://www.c-r.com/s_008.pdf
- [2] <http://www.ntt.co.jp/mirai/organization/organization0305.html>
- [3] http://www.hitachi.co.jp/wirelessinfo/al_index.html
- [4] <http://www.isid.co.jp/news/2011/1004.html>

付録

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#define IN 1
#define OUT 4
#define T 6200
#define Rad 25.f

int GetRandom(int min,int max);

int main(int argc,char *argv[])
{
    int i,j,k,count=0,flag=0;//カウント用変数
    double
x[IN+OUT][T],y[IN+OUT][T],z[IN+OUT][T],r[IN
+OUT][T];//ノード用変数[NODE ID][Time]の構成
    double
x2[IN+OUT][T],y2[IN+OUT][T],z2[IN+OUT][T],y
3[IN+OUT][T],x3[IN+OUT][T],R1[IN+OUT][T],R
2[IN+OUT][T];//推定ノードと発信ノードの距離格
納用変数
    double px[T],py[T],sx[IN+OUT],sy[IN+OUT];
    double
range=T,h1[IN+OUT],h2[IN+OUT],Z1[IN+OUT],
Z2[IN+OUT],ax[IN+OUT],ay1[IN+OUT],ay2[IN+
OUT],a1[IN+OUT],a2[IN+OUT],prex=0.f,prey=0.f
,Y1,Y2;

    FILE *A1,*A2,*H1,*H2,*R;

    A1 = fopen("A1.txt", "a");
    A2 = fopen("A2.txt", "a");
    H1 = fopen("H1.txt", "a");
    H2 = fopen("H2.txt", "a");

    R = fopen("R.txt", "a");

    //変数の初期化
    for(i=0;i<IN+OUT;i++)
    {
        sx[i]=0.f;
        sy[i]=0.f;
        ax[i]=0.f;
        ay1[i]=0.f;
        ay2[i]=0.f;
        a1[i]=0.f;
        a2[i]=0.f;
        h1[i]=0.f;
        h2[i]=0.f;
        for(j=0;j<T;j++)
        {
            x[i][j]=0.f;
            y[i][j]=0.f;
            z[i][j]=0.f;
            r[i][j]=0.f;
            x2[IN+OUT][T]=0.f;
            y2[IN+OUT][T]=0.f;
            z2[IN+OUT][T]=0.f;
            y3[IN+OUT][T]=0.f;
            x3[IN+OUT][T]=0.f;
        }
    }
    for(i=0;i<T;i++)
    {
        px[i]=0.f;
        py[i]=0.f;
    }

    //屋内ノード配置
    for(i=0;i<T;i++)
```

```

{
    if(i==0)
        {
            x[0][i]=GetRandom(21,39);
            y[0][i]=GetRandom(21,39);
            z[0][i]=GetRandom(0,25);
        }
    else
        {
            x[0][i]=x[0][i-1];
            y[0][i]=y[0][i-1];
            z[0][i]=z[0][i-1];
        }
}

//屋外ノード配置及び移動
for(i=1;i<IN+OUT;i++)
{
    for(j=0;j<T;j++)
    {
        switch(i)
        {
            case 1://NODEID 1
                if(j==0)
                    {
                        //初期配置
                        x[i][j]=20.0;
                        y[i][j]=0.f;
                        z[i][j]=0.f;
                    }
                else
                    {
                        //進行方向指定
                        x[i][j]=x[i][j-1];
                        y[i][j]=y[i][j-1]+0.01;
                        z[i][j]=z[i][j-1];
                    }
                r[i][j]=GetRandom(-250,250);
            case 2://NODEID 2
                if(j==0)
                    {
                        //初期配置
                        x[i][j]=0.f;
                        y[i][j]=40.0;
                        z[i][j]=0.f;
                    }
                else
                    {
                        //進行方向指定
                        x[i][j]=x[i][j-1]+0.01;
                        y[i][j]=y[i][j-1];
                        z[i][j]=z[i][j-1];
                        r[i][j]=r[i][j-1];
                    }
                break;
            case 3://NODEID 3
                if(j==0)
                    {
                        //初期配置
                        x[i][j]=40.0;
                        y[i][j]=range/100-0.01;
                        z[i][j]=0.f;
                    }
                else
                    {
                        //進行方向指定
                        x[i][j]=x[i][j-1];
                    }
                r[i][j]=GetRandom(-250,250);
            }
        }
    }
}

```

```

        y[i][j]=y[i][j-1]-0.01; //y 軸方向
        z[i][j]=z[i][j-1];
        r[i][j]=r[i][j-1];
    }
    break;
case 4: //NODEID 4
    if(j==0)
    {
        //初期配置
        x[i][j]=range/100-0.01;
        y[i][j]=20.0;
        z[i][j]=0.£;
        r[i][j]=GetRandom(-350,350);
    }
    else
    {
        //進行方向指定
        x[i][j]=x[i][j-1]-0.01;
        y[i][j]=y[i][j-1];
        z[i][j]=z[i][j-1];
        r[i][j]=r[i][j-1];
    }
    break;
default:
    printf("error of switch¥n");
    break;
}

//通信範囲測定用距離取得
z2[i][j]=sqrt((z[0][j]-z[i][j])*(z[0][j]-z[i][j]));
if(i%2==1)
{
    //通信判定
    //推定ノードと発信ノードとの3次元距離が実通信距離以下になると通信を開始
    if(R1[i][j]<=Rad+r[i][j]/100)
    {

```

```

if(j!=0&&j!=T-1)//初期侵入及び最
終侵入の判定
{
if(count==0)//侵入時二次元
半径取得
{
a1[i]=R1[i][j];//3次元距
離//RH
a2[i]=R2[i][j];//2次元距
離//r
}
//GPSデータの取得及び記憶
if(i%2==1)//Y軸方向
{
py[count]=y[i][j];
count++;
}
else//X軸方向
{
px[count]=x[i][j];
count++;
}
}
else//初期侵入及び最終侵入の除
去
{
sx[i]=100.0;
sy[i]=100.0;
break;
}
}
}
//推定2次元座標の算出
if(i%2==1&&py[count-1]>py[0])sy[i]=py[count-1]-
py[count-1]-py[0])/2;
else
if(i%2==1&&py[count-1]<py[0])sy[i]=py[0]-(py[0]-
py[count-1])/2;
else
if(i%2==0&&px[count-1]>px[0])sx[i]=px[count-1]-
(px[count-1]-px[0])/2;
else
if(i%2==0&&px[count-1]<px[0])sx[i]=px[0]-
(px[0]-px[count-1])/2;
else
{
sx[i]=100.0;
sy[i]=100.0;
}
}
//初回通信時発信ノード座標と推定2次元中間座標との距離
ax[i]=sqrt((py[count-1]-py[0])*(py[count-1]-py[0])/
2)//x
//推定ノードの推定2次元座標と推定2次元中間座標との距離
ay1[i]=sqrt((a1[i]*a1[i])-(ax[i]*ax[i]))//Y
ay2[i]=sqrt((a2[i]*a2[i])-(ax[i]*ax[i]))//y
}
else
{
//初回通信時発信ノード座標と推定2次元中間座標との距離
ax[i]=sqrt((px[count-1]-px[0])*(px[count-1]-px[0])/
2;

```

```

//推定ノードの推定 2次元座標と推定 2次元中間座標との距離
    ay1[i]=sqrt((a1[i]*a1[i])-(ax[i]*ax[i]));
    ay2[i]=sqrt((a2[i]*a2[i])-(ax[i]*ax[i]));
}
//記憶メモリの初期化
for(k=0;k<T;k++)
{
    px[k]=0.0;
    py[k]=0.0;
}
count=0;
flag=0;
}

count=0;

for(i=1;i<IN+OUT;i++)
{
    if(i%2==0&&sx[i]!=100.0)
        prex=sx[i];
    else if(i%2==1&&sy[i]!=100.0)
        prey=sy[i];
    else
        count++;
}

if(count<=2)
{
    for(i=1;i<=OUT;i++)
    {
        if(sx[i]!=100.f&&sy[i]!=100.f)
        {
            j=prex;
            Y1=((Rad)*(Rad)-ax[i]*ax[i])*((Rad)*(Rad)-ax[i]*ax[i]);
            Y2=sqrt(((Rad+r[i][j]/100)*(Rad+r[i][j]/100)-ax[i]*ax[i])*((Rad+r[i][j]/100)*(Rad+r[i][j]/100)-ax[i]*ax[i]));
            h1[i]=sqrt((-ay2[i]+sqrt(Y1))*(-ay2[i]+sqrt(Y1)));
            h2[i]=sqrt((-ay2[i]+sqrt(Y2))*(-ay2[i]+sqrt(Y2)));
        }

        printf("%f %f %f %f %f¥n",Rad,Rad+r[i][j]/100,h2,h1,r[i][j]/100);
        fprintf(A1,"%f¥n",Rad);
        fprintf(A2,"%f¥n",Rad+r[i][j]/100);
        fprintf(H1,"%f¥n",h1[i]);
        fprintf(H2,"%f¥n",h2[i]);
        fprintf(R,"%f¥n",r[i][j]/100);
    }
    fclose(A1);
    fclose(A2);
    fclose(H1);
    fclose(H2);
    fclose(R);

    return 0;
}

int GetRandom(int min,int max)
{
    static int flag;
    if (flag == 0)
    {
        srand((unsigned int)time(NULL));
        flag = 1;
    }
    return min + (int)(rand()*(max-min+1.0)/(1.0+RAND_MAX));
}

```