

アドホックネットワークを用いた
屋内位置の検出

龍谷大学 理工学部 情報メディア学科

学籍番号 T060569 小嶋 紘二

指導教員 三好 力 教授

内容梗概

近年、多くの大型ショッピングモールが建設されている。これらは、大規模な施設であり、館内入り口付近に設置してある案内板のみでは施設における現在地がわかりにくい。

現在では、無線 LAN を用いて、電波の到達時間差や電波強度から三辺測量により屋内の位置を検索するシステムがあるが、これは費用が最低で 500 万円と高価である。原因として、高精度の基地局の設置があげられる。そこで、アドホックネットワークを用いて、安価な基地局のみを使用し、精度の高い位置検索システムの構築を試みる。

第一章においては研究背景について述べる。

第二章では屋内位置検索とアドホックネットワークの概要、活用法について述べる。

第三章では、屋内ナビゲーションシステム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、今後の課題を述べ本論文の構成とする。

目次

第一章 はじめに
研究背景

第二章 既存技術
2.1 屋内位置検索
2.2 問題点
2.3 アドホックネットワークとは

第三章 提案手法
3.1 問題点
3.2 改良点
3.3 提案手法

第四章 実験方法と目的、結果
4.1 実験概要
4.2 実験目的
4.3 実験方法
4.4 実験環境
4.5 実験結果
4.6 考察

第五章 結論・今後の課題

おわりに

謝辞

参考文献

付録

第一章 研究背景

現在滋賀県では、多くの大型のショッピングモールが建設された。滋賀県においては、「琵琶湖クルージングモールピエリ守山」(137817平方メートル)があげられる。137817平方メートルというと東京ドームが建築面積46755平方メートルなので、東京ドーム約3つ分の敷地面積を持つこととなる。



図1 ピエリ縮尺図

さらに、2008年11月26日に敷地面積164961平方メートルをほこる「イオンモール草津」、2008年11月21日に「フォレオ大津一里山店」が55708平方メートル、2009年一月19日に「三井アウトレットパーク滋賀竜王」が敷地面積約180000平方メートルで相次いでオープンしている。



図2 イオンモール草津縮尺図



図3 フォレオ大津一理山縮尺図



図4 三井アウトレットパーク縮尺図

東京ドーム数個分の敷地面積を持つ上記施設の内部において、施設内の「どこに、なにがあるもか」ということが把握できなくなっている。そのような問題点を解決するために、「wifi ロケーションシステム」がある。このシステムについての概要は2章で解説する。

既存の屋内位置検出では、電波の到達の時間差で位置を判別する手法がある。この手法で位置を検出しようとする、高精度の時間同期と極めて短い時間を判断することができる基地局が必要となってくる。この基地局の設置が屋内位置検出に莫大な費用がかかる原因であり、現在、安価で実現できる、屋内位置検出システムが求められている。

第二章 既存技術

本章では、既存の技術の紹介を行いその問題点をあげ、本研究で用いるアドホックネットワークについて解説をおこなう。

2.1 屋内位置検索

2.1.1 移動機器連携ロケーショントラッキング方式

これは、検索の対象物を直接、高精度に推定しようとするのではなく、高精度に位置検索可能な移動機器を利用することで、間接的に対象物の位置を推定する方法である。

検索対象物に高価な専用デバイスを装備する必要をなくし、検索対象物の制限を緩和することが可能となる。また複数の位置検索可能な機器の移動と格機器が検知した情報をくみ合わせるにより位置推定制度の向上を図ることができる。

2.1.2 日立 AirLocation

日立 AirLocation は無線 LAN、IEEE802.11.b 準処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1～3m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

2.1.3 システム構成

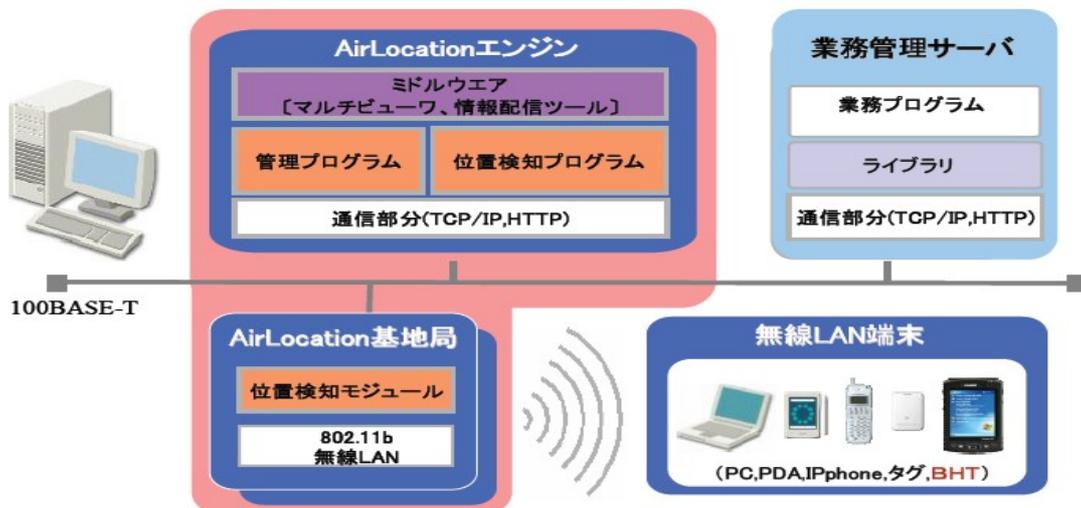


図5 システムの構成

図7のような基地局から端末に通信することによってその位置情報をおこなう。

2.1.4 日立 Location の測位方法

1) 2次元測位方式

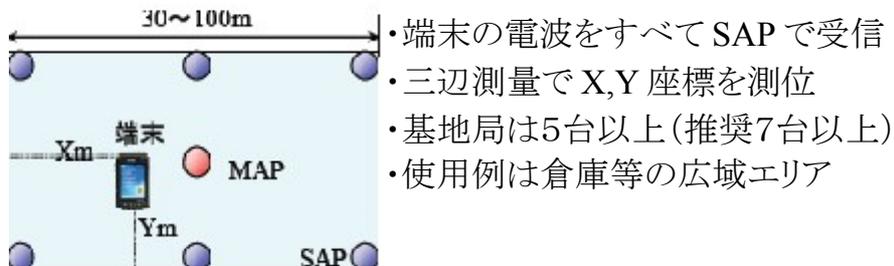


図6 2次元測位方式

2) 1次元測位方式

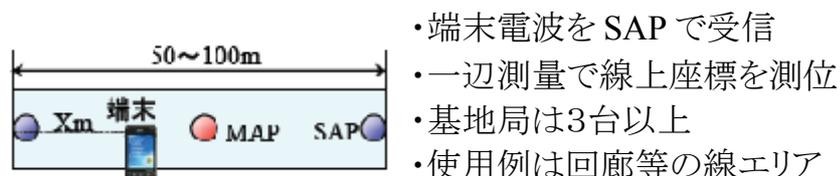


図7 1次元測位方式

3) アソシエーション方式

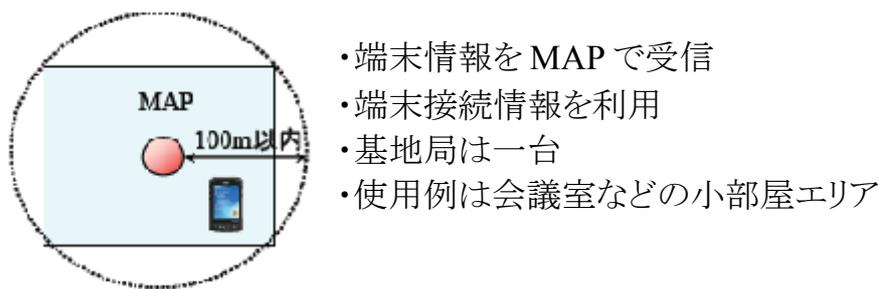


図8 アソシエーション方式

2.2 問題点

2.1 であげた屋内位置情報検索システムは2次元測位方式の場合、最小規格の場合でも「500万円」と高価なシステムとなっている。これは、基地局、位置測位の方法を電波の到達時間差などで判定してることから、高精度の基地局の時間同期が必要となってくるた

めである。

さらに精度向上のために多数の基地局を設置していることも問題である。この、高価なシステムを「安価に実現できないか」ということを目的とし、本研究では、「アドホックネットワークを用いた位置検出」の研究を行いシステムの構築をおこなう。

2.3 アドホックネットワークとは

アドホックネットワークとは基地局を用いずに、端末を利用してマルチホップ通信を行うネットワークのことを示す。従来の無線通信では、インターネットに接続した無線ルータやアクセスポイントに各端末が接続することで、端末同士が通信したり、インターネット上にあるサービスを利用することができる。

しかし、アドホックネットワークでは基地局を利用することなく、端末同士が直接通信し、その場でネットワークの構築を行う。アドホックネットワークの『アドホック』とは『その場限りの』という意味を持つ。

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。

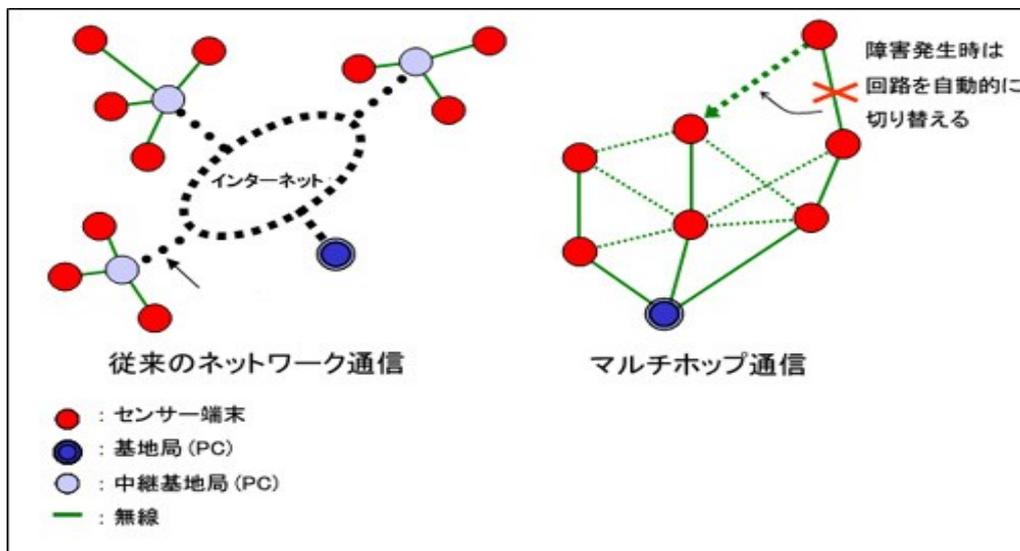


図9 アドホックネットワークによるマルチホップ通信

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 フラットティング

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トティングである。

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。

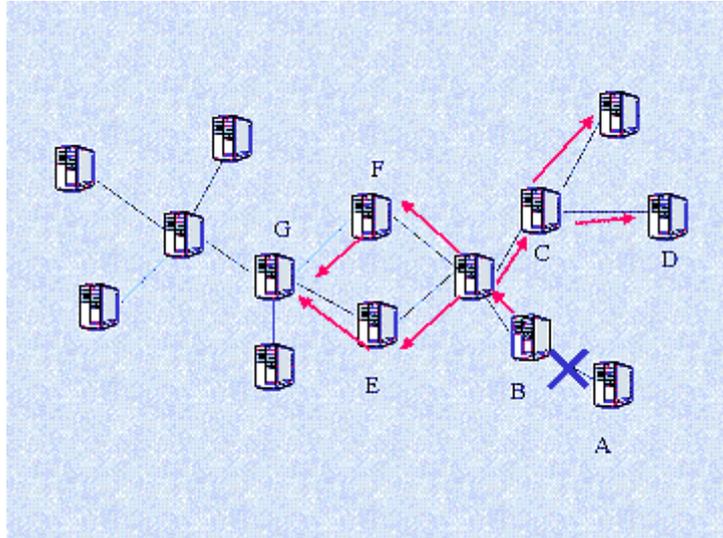


図 10 フラットティングの様子

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 2.3.3、2.3.4、2.3.5 に示す。

2.3.3 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。

2.3.4 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。

2.3.5 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おこなう型である。

第3章 提案手法

3.1 問題点

屋内位置検索システムは2次元測位の場合最小規格でも「500万円」と、大変高額なシステムとなっている。これは、位置検索を「電波の到達速度」と「電波強度」でおこなっているため、高精度に時間同期された高価な基地局が必要となってくる。さらに、位置検出の精度をあげるために、高価な基地局をさらに増やしている。

以上に挙げた

- 1) 高価な基地局の設置
- 2) 基地局の数

が主な問題点だと考えられる。

3.2 改良点

本研究では、高価な基地局を用いず、さらに基地局の数を減らすことを目的とする。

3.3 提案手法

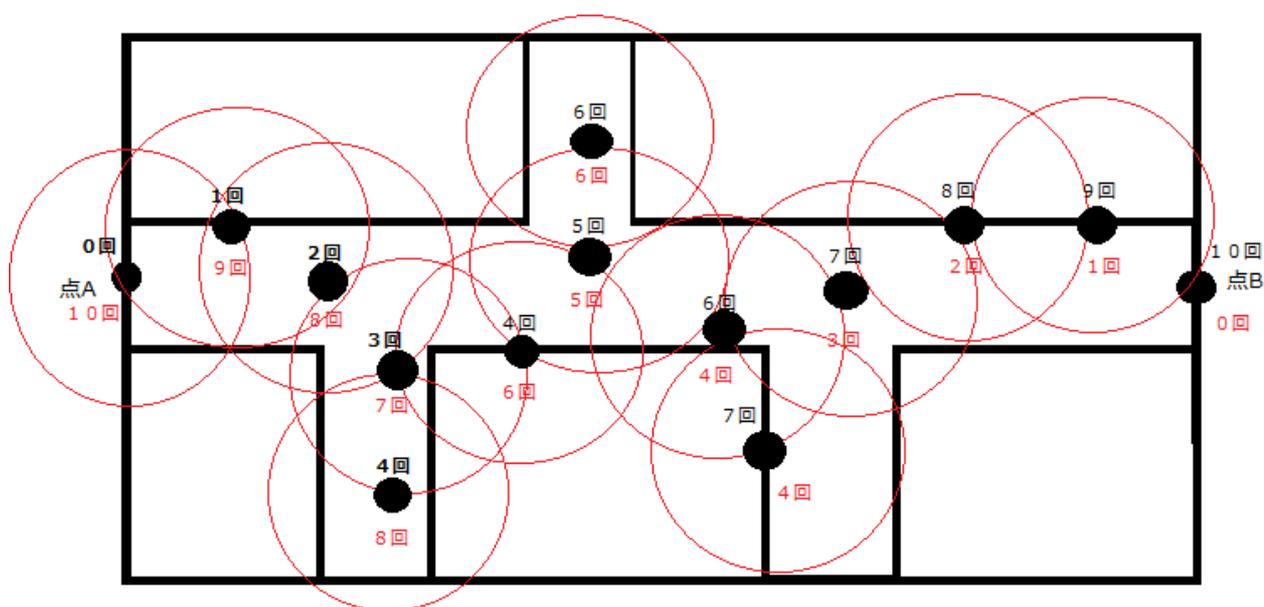


図 11 ポップ数の計測

図 11 のようにある屋内において点の位置を検索する場合の手法として、ポップ回数による判定をおこなう。図 7 の点 A 点 B からのポップ回数をカウントする。ポップ回数のため、直線距離とは違った値が出るが、その値の誤差は 2 点から同等に出ると仮定し、求めたポップ回数の比を用いて、位置情報をもった基地局の位置情報と通路地図から、目的の

端末の位置を検出する。たとえば、ポップ回数が A (0.25) から 5 回ポップ B (100.25) から 7 回ポップしたとすると、地図情報で道に指定されている場所 (図 12 斜線部) にしか点は存在しないことを利用すると

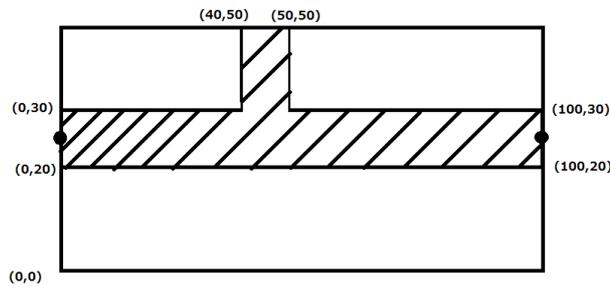


図 12 道指定の地図情報

$$x=(0-100)*(5/(5+7))$$

$$y=(25-25)*(5/(5+7))$$

で求められることがわかる。

3.4 アルゴリズム

1. 初期化

1. 1) ノードの数、TTL 値、通信半径の設定

2. すべてのノードをランダムに配置

3. 送信元ノードとあて先ノードを設定

4. フラッティング

4. 1) 送信元ノードは送信半径内にいるノードに対してパケットを送信

4. 2) TTL が設定値になった場合フラッティングを終了

4. 3) 送信したパケットと同じものを受信した場合、パケットを破棄

4. 4) パケットを受信したノードは、自分があて先でなかった場合、受信したパケットをそのパケットを送信したノード以外に送信半径内にいるノードに対して送信

4. 5) 受信したパケットが自分があて先だった場合次へ進む

4. 6) TTL を1増やす

4. 7) 4. 2)にもどる

5. 最短の経路の発見をおこなう

6. 送信先ノードを変更し4に戻る

7. 設置した基地局分繰り返し、すべて終了したら次へ進む

7. 固定地点からのポップ回数を用いて位置を算出する。さらに地図情報の道の位置を用いて計算をおこなう

7. 1) $40 < x < 50$ でないとき、 y の値は $y=25$ とし、 x の値は比を用い算出をおこなう。

7. 2) $40 < x < 50$ のとき、 x の値は同じく比を用いて算出し、 y の値はポップ回数と通信半径より算出をおこなう

第四章 実験方法と目的

4.1 実験概要

施設内に位置情報を持った基地局を設置し、そこに 50 個～300 個の人に当たる端末を配置する。位置情報を持った基地局から人に当たる端末にデータの送信を行い、位置情報の推定をおこなう。

さらに人に当たる端末同士でも通信を行い、位置情報を持った基地局と通信ができない場合でも位置を推定できるようにする。

位置情報を持った基地局を 3 個設置し実験をおこなう。実際に配置した人に当たる端末との誤差の判定を行い実用範囲内のレベルを算出する。

4.2 目的

実験をおこない、実際の点の座標と求められた点の座標を比べ、本手法が有効かの検討をおこなう。

はじめに、グラフ化を行い、視覚的にどの程度の誤差が生まれるかを判断し、次に 2 点間の差を出し、標準偏差を求め、判定をおこなう。

4.3 実験方法

本実験では、ランダムに打たれた点の実際の位置と提案手法で求められた位置との差を比較評価するため、c 言語でのシミュレーションをおこなった。

4.4 システム概要

- 1、図 12 を用い通路にのみランダムに点を配置する。
- 2、固定点を図 13 の黒丸に 3 箇所設け、固定点から通信可能範囲に対して通信をおこなう。
- 3、通信した回数をポップ回数とし、カウントをおこなう。
- 4、さらに通信した先の端末からも順々に通信をおこなっていく。
- 5、目的地にたどりついたら、通信を終了する。
- 6、最短のルート検索を行いその値をポップ回数とする。
- 7、設置した基地局分のポップ回数から距離の判定をおこなう。

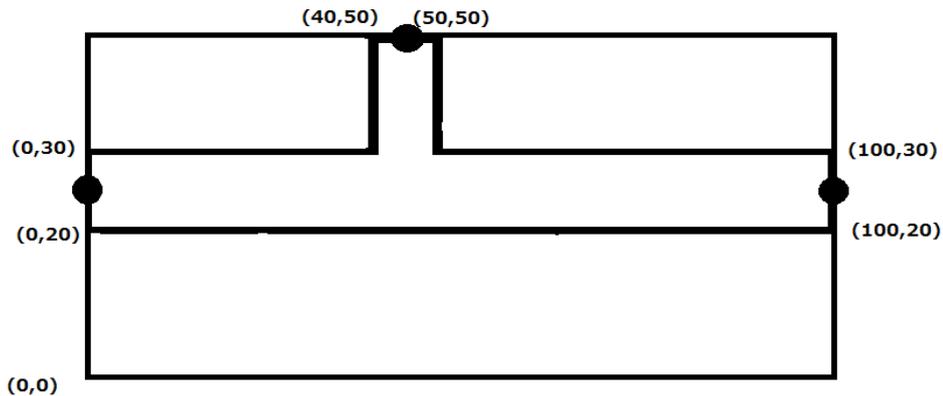


図 13 基地局の設置箇所

4.5 実験環境

実験環境は実際の施設のように障害物のある空間とし 100×50 の空間において、図 12 のような通路においてのみの実験とする。

フィールドに 50 個から 300 個のノードをそれぞれランダムに配置させる。各ノードは通信半径以内に存在するノードと直接通信することができ、マルチホップ通信を用いてアドホックネットワークを形成する。パラメーターごとに 100 回の試行を行い、シミュレーション上で配置した位置と、計算上の値の比較をおこなう。詳しいパラメータは表 1 にまとめる。

表 1 シミュレーションパラメータ

モデルサイズ	100×50
ノード数	50,100,300,
通信半径	5,10,20
試行回数	100

4.6 実験結果

実験結果を示す。

表 2～6 は、算出した値の差を計算し、標準誤差を求めたもの、図 14～18 は、実際の位置を青■であらわし、求めた位置を赤◆で表示したものである。

ノード数 50、半径 5 の結果を表 2、図 14 に、ノード数 100、半径 5 の結果を表 3、図 15 に、ノード数 300、半径 5 の結果を表 4、図 16 に、ノード数 100、半径 10 の結果を表 5、図 17 に、ノード数 100、半径 20 の結果を表 6、図 18 に示す。

表2 標準誤差(ノード50、半径5)

平均	7.88	3.28
標準偏差	4.07	4.86
サンプル数	50	50
標準誤差	0.81	0.97

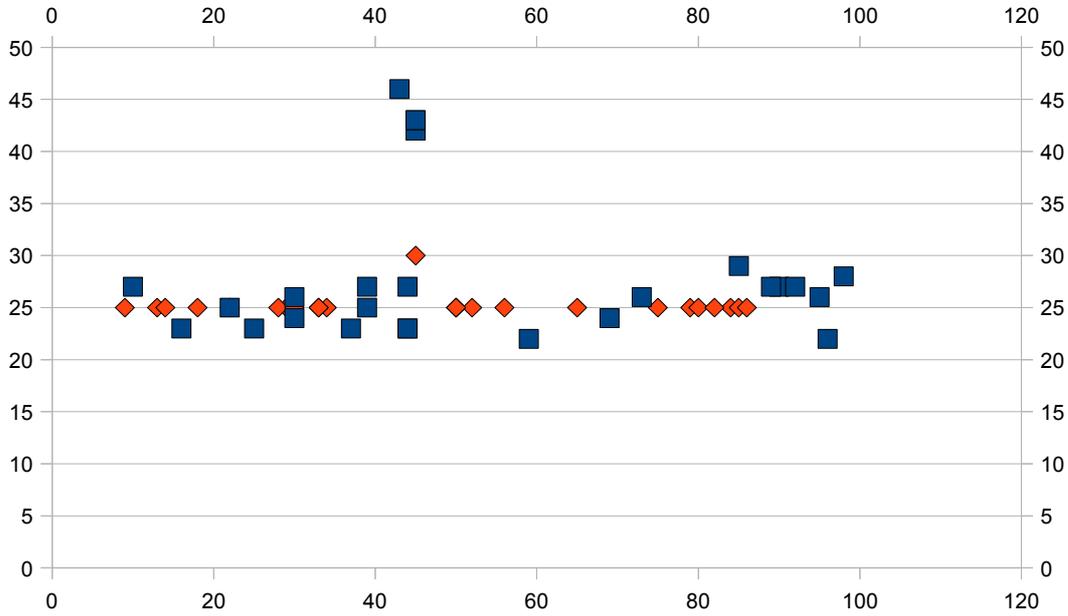


図14 2種類の点の座標の関係(ノード50、半径5)

表3 標準誤差(ノード100、半径5)

平均	6.92	4.16
標準偏差	3.89	5.52
サンプル数	100	100
標準誤差	0.55	0.78

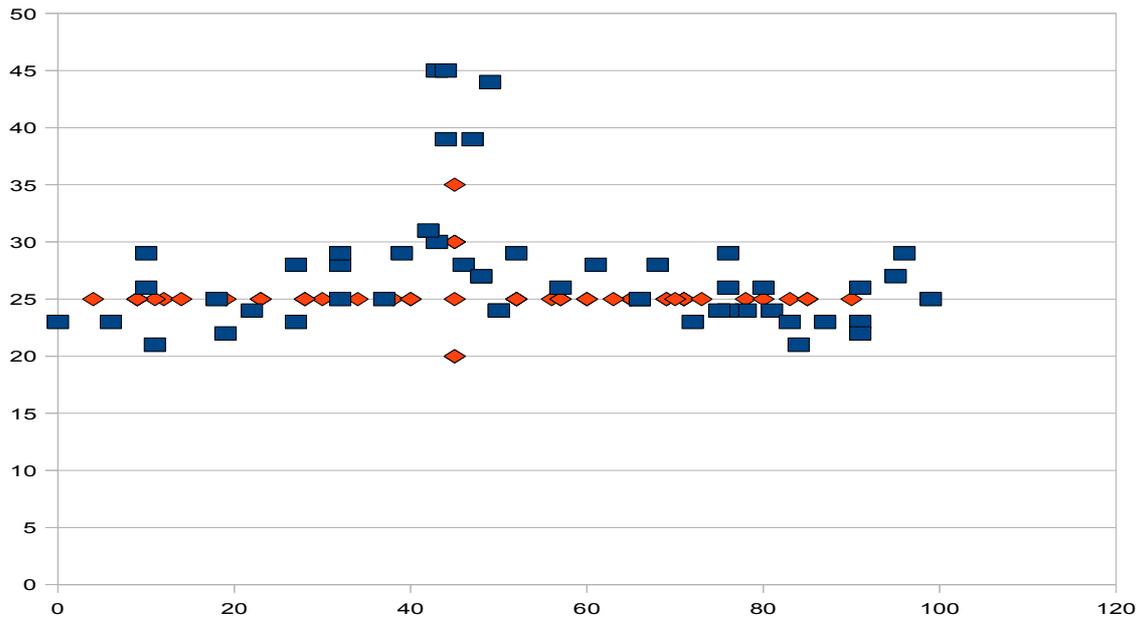


図 15 2種類の点の座標の関係(ノード 100、半径 5)

表 4 標準誤差(ノード 300、半径 5)

平均	7.34	3.41
標準偏差	3.94	4.13
サンプル数	300	300
標準誤差	0.32	0.34

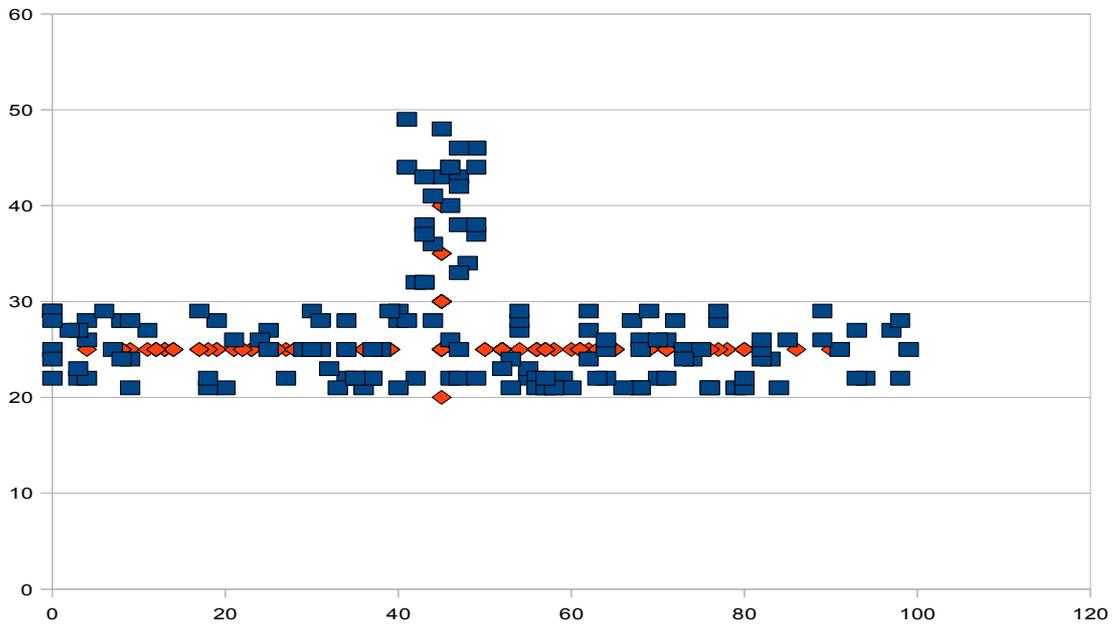


図 16 2種類の点の座標の関係(ノード 300、半径 5)

表 5 標準誤差(ノード 100、半径 10)

平均	6.66	5.56
標準偏差	4.25	6.75
サンプル数	100	100
標準誤差	0.6	0.95

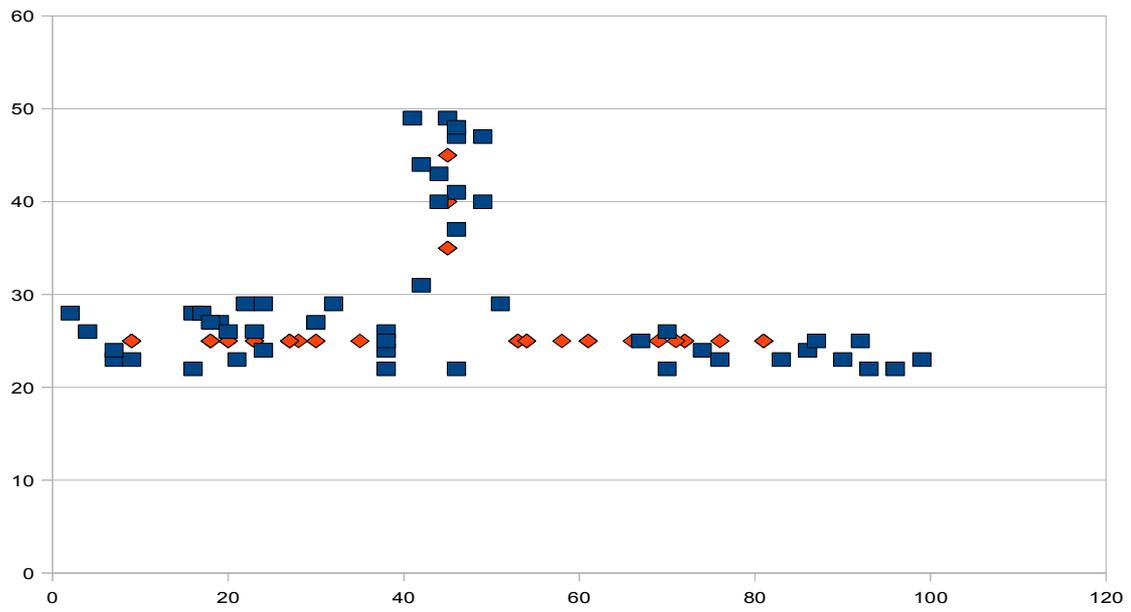


図 17 2 種類の点の座標の関係(ノード 100、半径 10)

表 6 標準誤差(ノード 100、半径 20)

平均	9.38	4.76
標準偏差	5.93	5.39
サンプル数	100	100
標準誤差	0.84	0.76

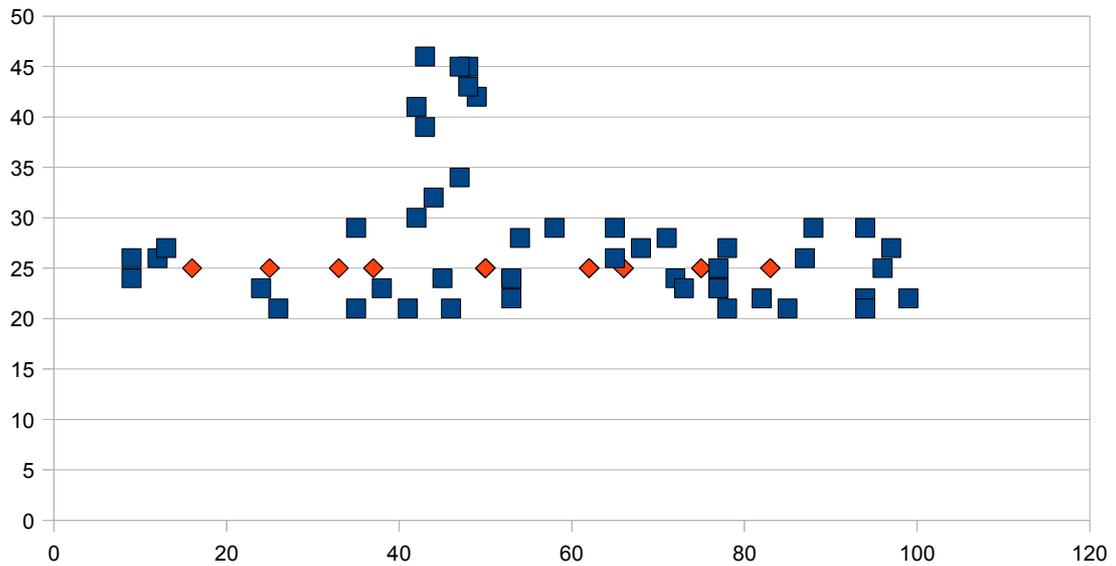


図 18 2 種類の点の座標の関係(ノード 100、半径 20)

4.7 考察

図 14 図 15 図 16 図 17 図 18 は図 12 の擬似施設内に、ランダムに点を打った状態を表している。青■は、乱数を発生させシミュレーション上で打った点、赤◆はランダムに打った点を元に定点からのポップ回数のカウントを行い、地図情報を用いて計算したものである。

表 2(ノード数 50 半径 5)の結果と表 3(ノード数 100 半径 5)の結果、表 4(ノード数 300 半径 5)の結果を比べてみると、標準誤差が表 2 では(0.81,0.97)となっているのに対し表 3 では(0.55,0.78)、表 4 では(0.32,0.34)とノード数を増やすごとに誤差の少ない値が出ていることがわかる。

次に表 3(ノード数 100 半径 5)、表 5(ノード数 100 半径 10)、表 6(ノード数 100 半径 20)で標準誤差を比べてみると順に(0.55,0.78)(0.60,0.95)(0.84,0.76)とこちらは半径が小さいほど誤差の少ない結果となっている。

さらに、ノード数 300 半径 5 のとき既存技術に近い実験結果が確認された。

第五章 結論

本論文では、既存の屋内位置算出システムの問題点を挙げ、高価な基地局を用いず、さらに基地局の台数を減らすことができる、アドホックネットワークを用いた屋内位置算出システムを提案した。

実験結果から、ノード数が多い状況で、さらに通信半径を小さくすることにより既存の位置検索算出システムに匹敵できることが確認できた。限られた状況ではあるが、本研究で提示した提案手法は、既存の位置算出システムと比べ、有効だと確認できた。

今後の課題として、状況を選ばずに、既存のシステムを超えるものの開発をおこなう。

謝辞

本研究を進めるにあたり、ご指導いただいた龍谷大学情報メディア学科の三好力教授に深く感謝いたします。また、日常の議論を通じて多くの知識や示唆を頂いた同研究室の皆様や多大なご協力を頂いた皆様に深く感謝いたします。

最後に、私を支えてくれた家族・友人に深く感謝します。

参考文献

[1]「日立 AirLocation と位置情報ソリューション」

http://www.symbol.co.jp/pdf/seminar/6_HITACHI.pdf

[2]未来ネット研究所

<http://www.ntt.co.jp/mirai/organization/organization0305.html>

[3]シナモニ

http://www.shinamoni.jp/web/technology/monitoring_multihop.html

付録

<ポップ回数のカウント>

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#define SQUx 100 //施設の範囲
#define SQUy 50
#define SECOND 1000//試行回数
#define RENG 10//通信半径
#define MAX 300//ノードの総数
#define initialx1 0//基地局座標x
#define initialy1 25//基地局座標y
#define initialx2 100//基地局座標x
#define initialy2 25//基地局座標y
#define initialx3 45//基地局座標x
#define initialy3 50//基地局座標y
#define TTL 0 //ポップ回数
int main(void){
int yyyy=0;
while(yyyy!=3){
int nodex[MAX];//ランダムノードのx配列
int nodey[MAX];//ランダムノードのy配列
int kakunoux[MAX];
int kakunouy[MAX];
int POP[MAX];
int ln=1;
int MIN = SECOND;
int rupehojix[MAX];
int rupehojiy[MAX];
int ihoji[MAX];
int rengx;
int rengy;
int keisan[MAX];
FILE *fp;
fp = fopen("loop.txt","a+");
int i , j;
int TTL =0;
for(i =0; i < MAX; i++){
kakunoux[i] = 0;
kakunouy[i] = 0;
POP[i]=SECOND;
}
srand((unsigned) time(NULL));
int p =0;
while(p != MAX){
int c = rand() % SQUx;
int d = rand() % SQUy;
if( (20<d && d<30) || ((20<c && c<30) && (0<d && d<30)) || ((40<c && c<50) && (20<d && d<50)) ) {
nodex[p]=c;
nodey[p]=d;
p++;
}}
int xyz = rand() % MAX;
int rupekaisu=1;
int ky=0;
for(ky=0;ky<3;ky++){//(nodex[1].nodey[1])
while(rupekaisu != MAX){
TTL = 0;
rupekaisu++;
for(i =0 ; i < MAX ; i++){
if(ky==0){
```

```

rengx = (initialx1-nodex[i]) * (initialx1-nodex[i]);
rengy = (initialy1-nodey[i]) * (initialy1-nodey[i]);
}
if(ky==1){
rengx = (initialx2-nodex[i]) * (initialx2-nodex[i]);
rengy = (initialy2-nodey[i]) * (initialy2-nodey[i]);
}
if(ky==2){
rengx = (initialx3-nodex[i]) * (initialx2-nodex[i]);
rengy = (initialy3-nodey[i]) * (initialy2-nodey[i]);
}
int reng = RENG * RENG;
int rengxy =rengx +rengy;
if(reng > rengxy){
if(initialx1==nodex[i] && initialy1 == nodey[i]){break;}
else {
if(xyz == i){
ln++;
POP[ln]=TTL;
}
else{
rupehojix[TTL] = nodex[i];
rupehojiy[TTL] = nodey[i];
TTL = TTL + 1;
while(TTL != 10){
int k;
for( k = 0; k < MAX; k++){
int rengx = (rupehojix[TTL] - nodex[k]) * (rupehojix[TTL] - nodex[k]);
int rengy = (rupehojiy[TTL] - nodey[k]) * (rupehojiy[TTL] - nodey[k]);
int reng = RENG * RENG;
int rengxy = rengx + rengy;
if(reng > rengxy){
if((rengx == 0 && rengy == 0) ||(rupehojix[TTL-1]== nodex[k])&&(rupehojiy[TTL-1]==nodey[k]) ||(rupehojix[TTL-2]==
nodex[k])&&(rupehojiy[TTL-2]==nodey[k])||(rupehojix[TTL-3]== nodex[k])&&(rupehojiy[TTL-3]==nodey[k])){
}
else {
if(xyz == k){
ln++;
POP[ln]=TTL;
break;
}
else{
TTL = TTL + 1;
rupehojix[TTL] = nodex[k];
rupehojiy[TTL] = nodey[k];
}}}}
break;
}}}}
int rup;
for(rup=1;rup<MAX; rup++){
if(POP[rup]<MIN){
MIN = POP[rup];
POP[rup]=SECOND;
}}
fprintf(fp,"%d,",MIN+1);
keisan[ky]=MIN+1;
MIN=SECOND;
fprintf(fp,"%d,%d,\n",nodex[xyz],nodey[xyz]);
fclose(fp);
yyyy++;
}
return 0;
}

```

```

<ポップ回数から比で計算>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#define SQUx 100 //施設の範囲
#define SQUy 50
#define SECOND 300//試行回数
#define RENG 20//通信半径
#define MAX 100//ノードの総数
#define initialx1 0//基地局座標x
#define initialy1 25//基地局座標y
#define initialx2 100//基地局座標x
#define initialy2 25//基地局座標y
#define initialx3 45//基地局座標x
#define initialy3 50//基地局座標y
#include <stdio.h>
int main(void){
int xm,ym,k1,k2,k3;
FILE *file1;
FILE *file2;
int k=0;
int l=0;
int qx,qy;
int x,y;
file1 = fopen("loop.txt","r+");
file2 = fopen("ratio.txt","w+");
while(k != SECOND){
fscanf(file1,"%d,%d,%d,%d,%d",&xm,&ym,&k1,&k2,&k3);
x=SQUx*k1/(k1+k2);
if(40<x && x<50){
qx=45;
qy=SQUy-(k3*RENG);
}
else{
qx=x;
qy=25;
}
fprintf(file2,"%d,%d,%d,%d\n",xm,ym,qx,qy);
k++;
}
fclose(file1);
fclose(file2);
return 0;
}

```