

携帯端末の位置 探索手法の検討

龍谷大学 理工学部 情報メディア学科
学籍番号 T060589 辻 智弥
指導教員 三好 力 教授

内容梗概

東日本大震災によって、多くの建物が広域にわたり倒壊し、逃げ遅れた人々が瓦礫の下敷きになり命を落とした。このような震災が起こったとき、瓦礫に埋もれた人々をいかに早く助け出すことができるかが重要である。

広い範囲での被害となると、すべての瓦礫を除去することは非常に困難であり、また人が瓦礫に埋もれてしまった場合 72 時間以内に助け出さなければ生存率は激減するため、できるだけ人がいるポイントを中心に除去作業をする必要がある。そのためには何らかの手法を使い人がいる位置を特定しなければならない。

本研究ではこの問題を解決するために、人が普段身につけている携帯電話を用いて人を見つけ出すことを目標とし、シミュレーションを使い発見できるかどうかを調べた。シミュレーションでの実験の結果、携帯電話の密度が高い場合は推定範囲が広く場所特定は困難であったが、携帯電話の密度が低い場合、場所の推定も行うことができた。

目次

1. はじめに
 - 1.1 本研究の背景
 - 1.2 本研究の目的
2. 既存技術
 - 2.1 スカイメッシュ
 - 2.2 スカイメッシュを利用した位置探索手法
 - 2.3 緊急公助ワイヤレスソリューション (NEC)
 - 2.4 問題点
3. 提案手法
4. 実験
 - 4.1 実験 1
 - 4.2 実験 2
5. 実験結果・考察
 - 5.1 実験 1 の結果
 - 5.2 実験 2 の結果
 - 5.3 考察
6. まとめ

1 はじめに

1.1 本研究の背景

2011年3月11日に発生した東日本大震災は図1のように広範囲にわたり地震を発生させ、建物が崩壊し人が瓦礫の中に閉じ込められたり、津波によって流されるという事態が発生した。この地震によって平成24年10月24日時点での警察庁緊急災害警備本部の発表によると、人的被害は死者15872人、行方不明者2777人、建物被害においては全壊した建物は129577戸、半壊が266101戸という甚大な被害をもたらした。

また首都直下型でマグニチュード7以上の地震は30年以内に70%の確立で発生するという内閣府の報告や、東海・東南海地震などもいつ起こってもおかしくない状況になっており、大地震にそなえた対策が必要になってきている。

そのような地震が起こるとき、事前に地震を予知することも必要だがいくら予知しても大地震が起これば耐震構造が十分でない建物は倒壊する。耐震構造が不十分な建物をなくすことは不可能であり、もし仮にすべての建物に対策を練ることができたとしても、10年20年と年数を経つごとに老朽化していく。そのため大地震が発生した場合、被害が出ることは防ぐことができないが、その後の行動により被害を減らすことは可能である。

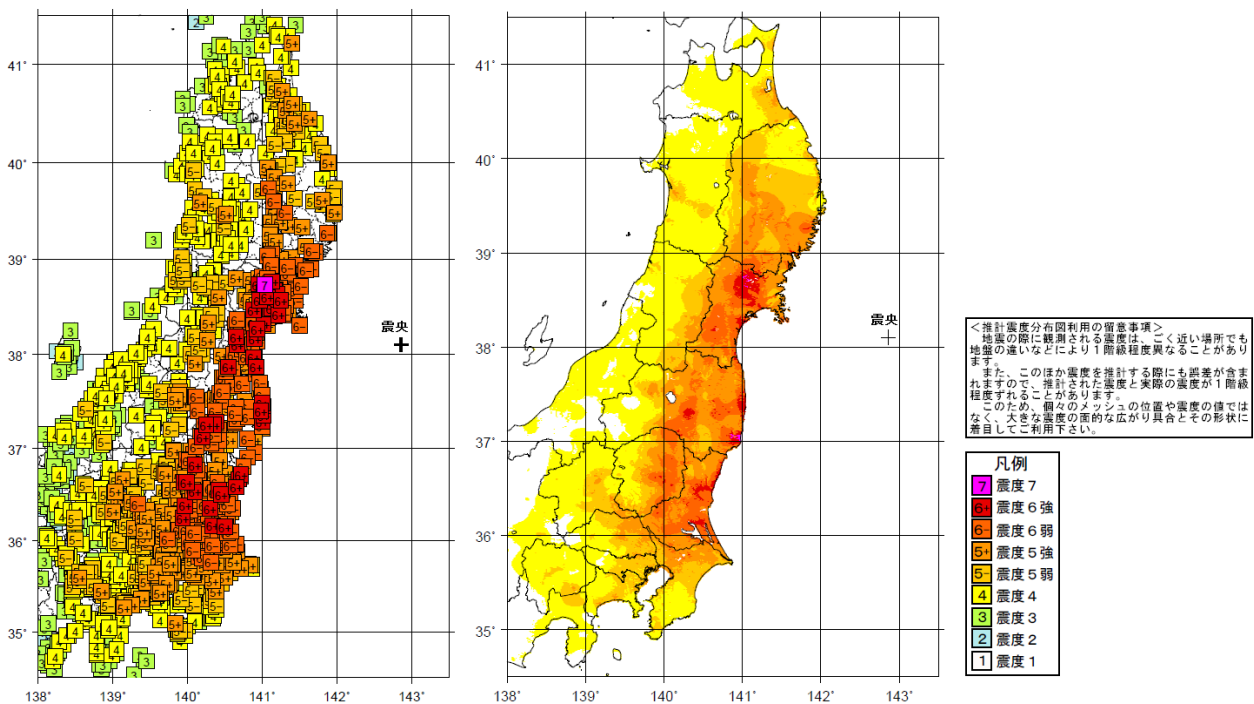


図1:東日本大震災の震度

建物の倒壊などによって瓦礫に埋もれてしまった場合、72時間以内に発見することができなければ脱水症状や低体温症などが発生しはじめ、生存率が極端に下がってしまう。そのため地震が発生し、建物の倒壊などが発生した場合、早期発見することが非常に重要になってくる。

1.2 本研究の目的

現在行われている倒壊建物内の探索では、広域災害など広範囲で発生した災害現場で全範囲をくまなく探索するのは時間も人手も効率がいい探索とはいえない。倒壊した建物の中から被災者を迅速に発見するためには、探索範囲を事前に絞ることで、少ない人手で効率のよい探索が行え、余った人手をほかの地域に派遣することで広範囲をカバーすることができる。

本研究では瓦礫に埋もれた人々を早期発見するという課題を達成するために、車の上に乗せた強い指向性を持ったアンテナを360度回転させながら走ることによって、移動しながら携帯電話の位置を特定することで、瓦礫に埋もれた人々の位置を特定することが目的である。

2 既存技術

2.1 スカイメッシュ

スカイメッシュとは気球を用いたアドホックネットワークである。気球にノード、アンテナなどをつけた簡易基地局を空に上げることで、通信環境を構築する。またカメラをつけることにより、周囲の撮影も同時に行え、災害時において円滑な救助活動を行うことを目的としている。

2.2 スカイメッシュを利用した位置探索手法

スカイメッシュによって通信環境を構築する時、各ノードの受信範囲をあえてある程度重ねることにより、複数のノードに接続する携帯電話の位置を重なり合った受信範囲を用いて探索する。また電波強度を用いることで重なり合った範囲からさらに範囲を絞ることができる。

2.3 緊急公助ワイヤレスソリューション (NEC)

バルーンやクレーンによって携帯電話のアンテナを吊り下げ、そこに対して受信してきた信号の位置を特定することによって遭難者の位置をわりだすというシステム。図2のように高度60m程度の高さで半径約10m程度の範囲を調べることができると推定されているが、2012年10月31日時点ではまだ実験を行っていないため範囲は不明である。

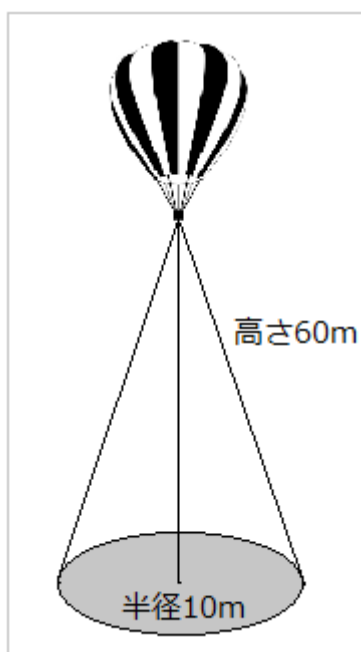


図2:緊急公助ワイヤレスソリューションの例

2.4 既存技術の問題点

既存手法はバルーンを用いているが、高度 10m 付近の風速に対して 50m 付近の風速は約 1.6 倍程度あり、風速が 8m 付近までなら安定するが、それ以上の風速がある時、動作が不安定になる場合もあり使用の為には気象条件に左右される。また多くのバルーンを揚げるためには、バルーンや浮力を得るためのヘリウムガスを大量に用意しなければならず、備蓄がない場合、広域災害の初期段階において用意し災害現場まで輸送するために多くの人手がとられてしまう。

またバルーン以外にクレーンを使う場合であっても、震災時に各地に大量のクレーンを配置することは難しく、また移動できる範囲も限定されることが予想される。

3 提案手法

本手法では既存技術の欠点を解決するために、図1のように車の上に乗せた強い指向性を持ったアンテナを360度回転させながら走ることによって、移動しながら携帯電話の位置を特定することを提案する。携帯電話の電波を受信した地点の位置情報とアンテナの方向を記録することで、多くの直線が地図上に記録される。その直線が交差する地点に携帯電話があると推定される。

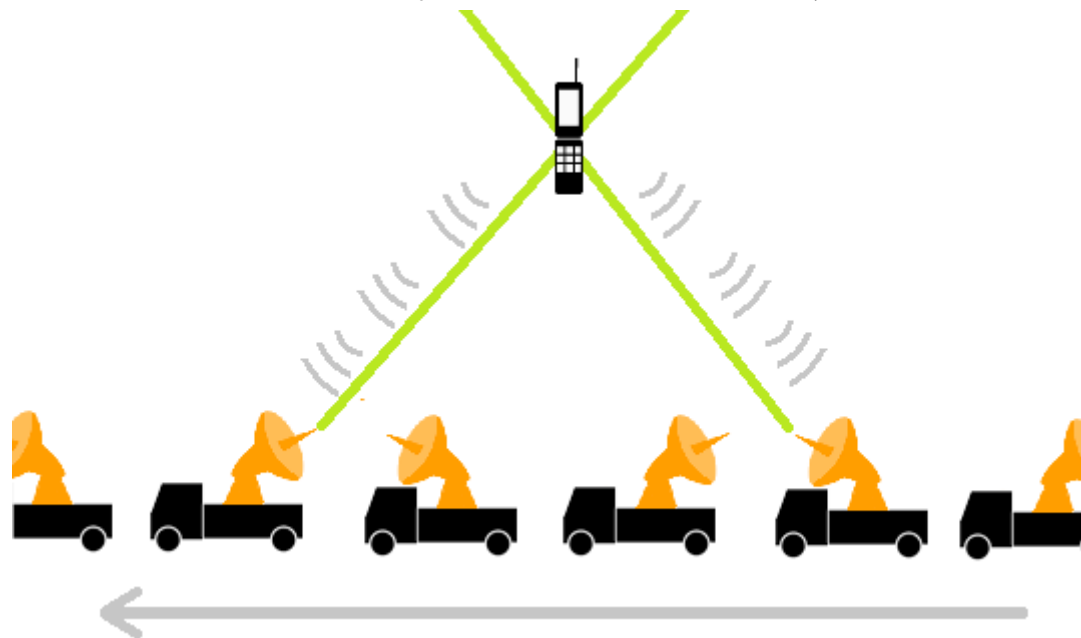


図3 提案手法の探索イメージ図

また、携帯電話の電波は第3世代携帯で約10km届くといわれているが、それは基地局にある受信アンテナの性能が良かったためであり、震災時にそのようなアンテナを用意することは難しい。携帯電話やスマートフォンが使用している電波の帯域は、800MHz,1.5GHz,2.0GHzなど、出力は200~250mWのものが一般的に普及している。またBluetoothの電磁波が2.4GHz、100mWと比較的近い数値のためBluetoothを参考にすると、Bluetoothの100mW送信機で約100mの通信が可能である。これより第三世代携帯電話端末の通信距離は約100~140m程度以内と推定することができる。しかし瓦礫などの下にあることを考えると140m届くことは考えづらい。これを踏まえて実験を行う。

4 実験

4.1 実験 1

実験 1 では、強い指向性を持つアンテナを自動車につみ、アンテナを回転させながら走行してデータを集めることを考えている。今回は図 4 のような直線の移動経路上を移動しながら携帯電話端末を探す事を考える。携帯電話からの電波が届く範囲は、瓦礫などにより妨害されていると仮定し、 $r=100\text{m}$ に設定した。位置測定に用いる方法は、電波を受信した方角と探索体の位置を記録していく。強い指向性のため、探索体からある角度を持った直線が引かれる。そのため図 5 のように複数の線が携帯電話に向けて伸びていくことがわかる。その線が交差するところに携帯電話がある、すなわち遭難者がいる可能性が高いということになる。そして、直線の交点と携帯電話の実位置とのずれを測定し、携帯電話の位置を正確に推定できるかどうかを確かめることが目標である。

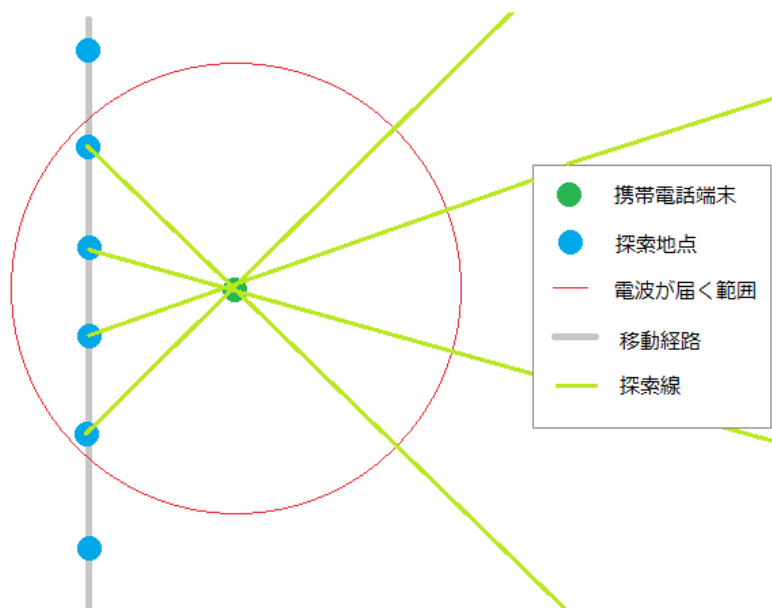


図 4 実験1の探索方法

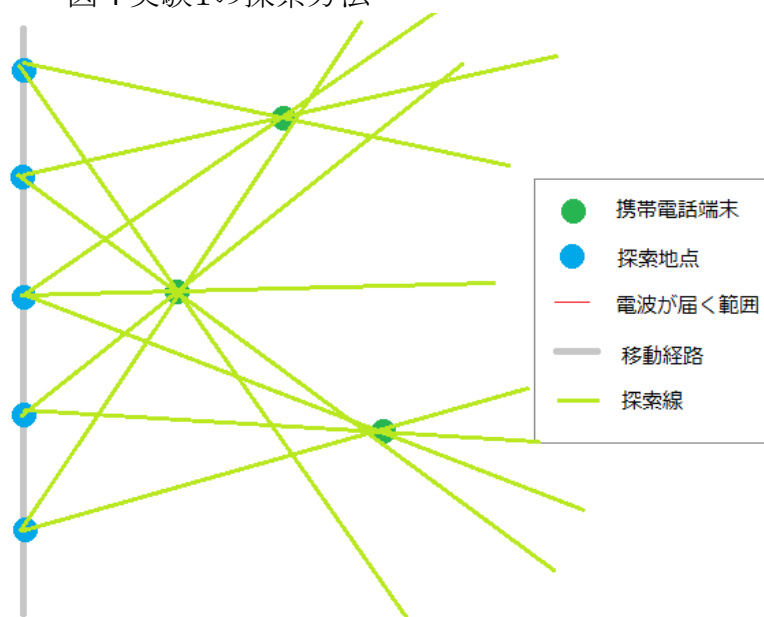


図 5 実験 1 の探索結果例

実験1では電波を受信する方向が1本の直線で測定できるという理想的な環境を想定することで、携帯電話の位置が特定できるかどうかを確認する。

実験の流れは以下の通りである。

- ①携帯の実位置をランダムに決定する
 - ②探索体を一定の速度で移動させる
 - ③探索体上で一定の角速度で直線を回転させる
 - ④携帯の実位置と直線が交差したらその時の探索体の位置と直線の角度を記録する
- 上記を繰り返しデータを収集する。

収集後地図上に記録した位置から記録した角度の直線を引き、直線の交点を携帯の推定位置とする。

図6のように、携帯がある箇所は多くの直線が交差する交点にあるので、多く重なった点を携帯の位置と推定する。

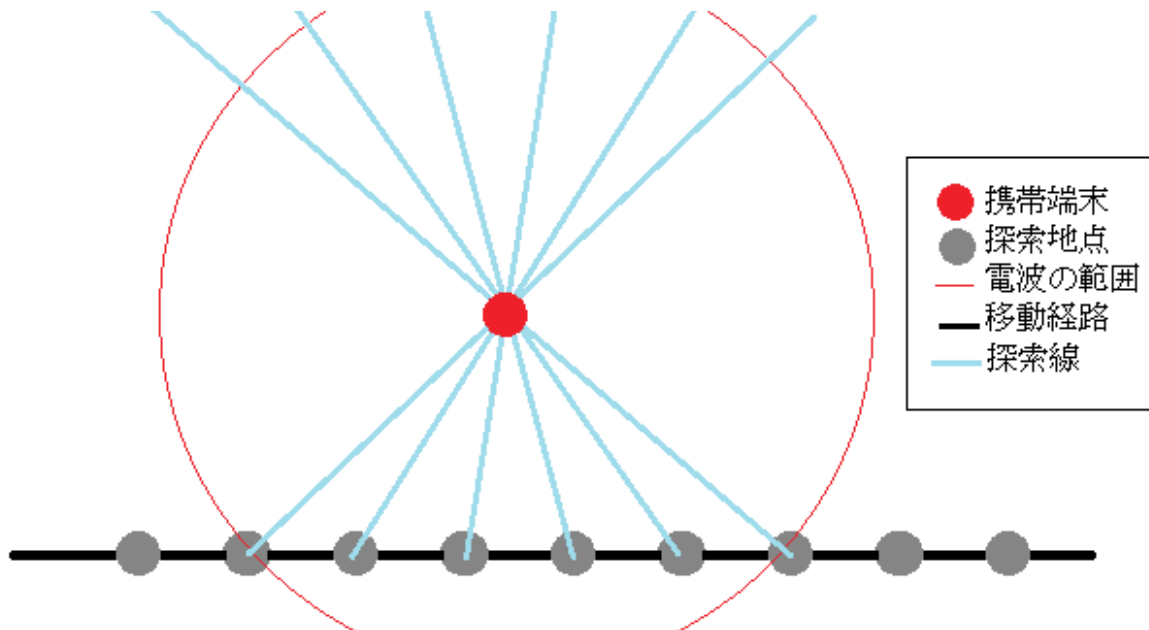


図6 実験1の探索方法と結果イメージ図

4.2 実験 2

実験 1 では探索体から伸びる直線状に携帯電話があると設定していた。これは強い指向性があるとしたためであるが、このような条件に当てはまるアンテナは現実には存在せず、通常はある一定の探知範囲をもち、その探知範囲内に携帯電話端末が存在しているという程度にしか測定することはできない。したがって実験 2 では図 7 のように探索体からある一定の角度を持つ探知範囲内に携帯電話端末の電波が届いた場合、その範囲内に携帯電話端末が存在すると記録させる。これを実験 1 のように直線上を走る探索体と一定の角速度のアンテナをつかい測定を行う。携帯電話端末が存在する場所の付近は複数回記録されるため、図 8 のように記録された回数に基づいた地図を作り、携帯電話の実位置と比べることにより携帯電話を発見することができるかどうかを測定する。

アンテナの指向性の強さと推定位置の関係を明らかにするため、実験 2 では実験 1 の④を以下とする。

④携帯の実位置と探知範囲が交差している間その時の探索体の位置とアンテナの角度を記録する

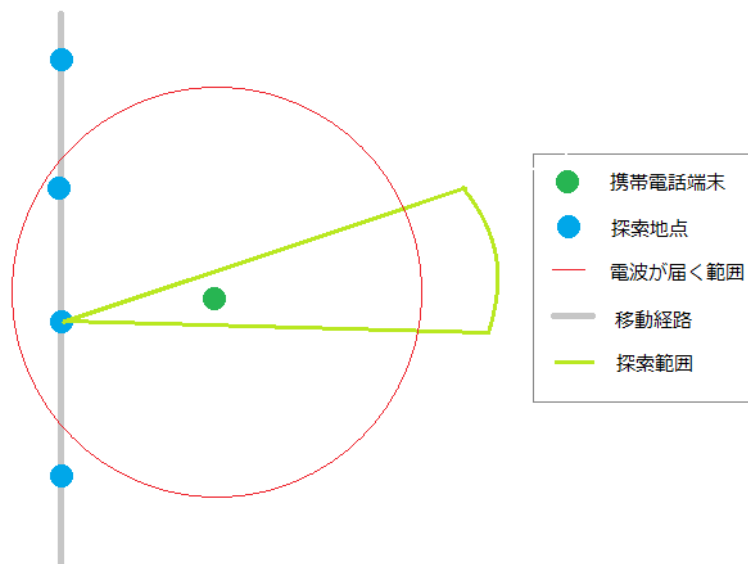


図 7 実験 2 の探索方法

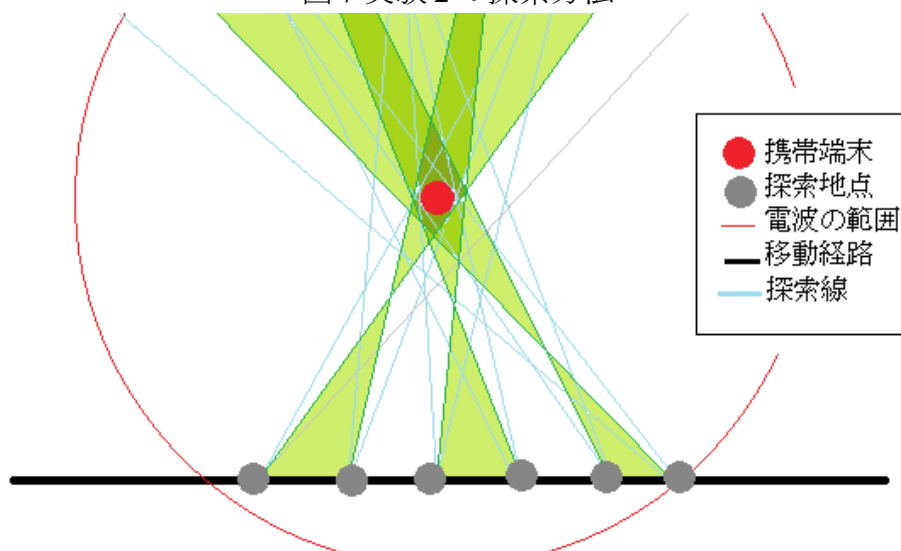


図 8 実験 2 の探索結果イメージ図

5 実験結果・考察

5.1 実験1の結果

表 1: 実験条件 1

探索領域	250×250
携帯電話数	10
通信半径	100
探索点の間隔	10

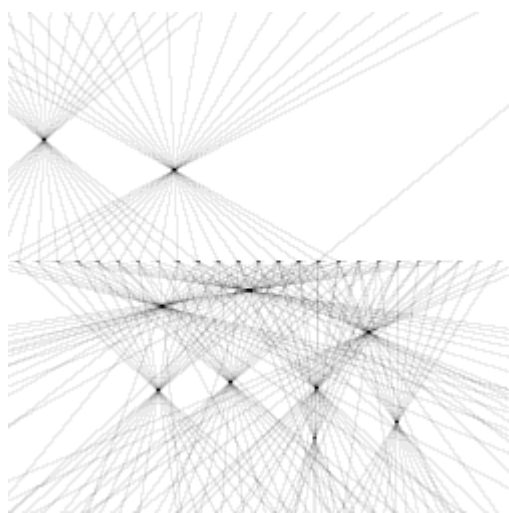


図 9.1 実験 1 の実験結果

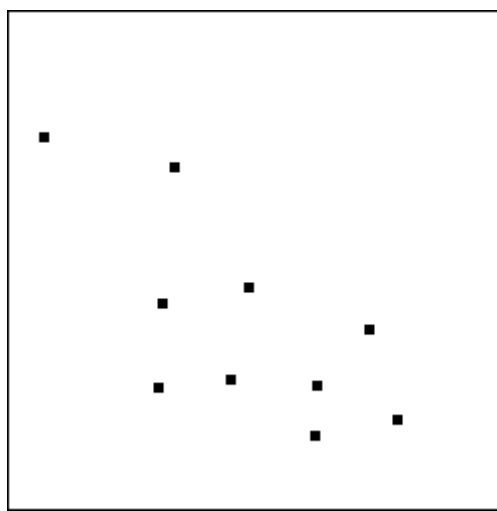


図 9.2 実験 1 の携帯電話実位置

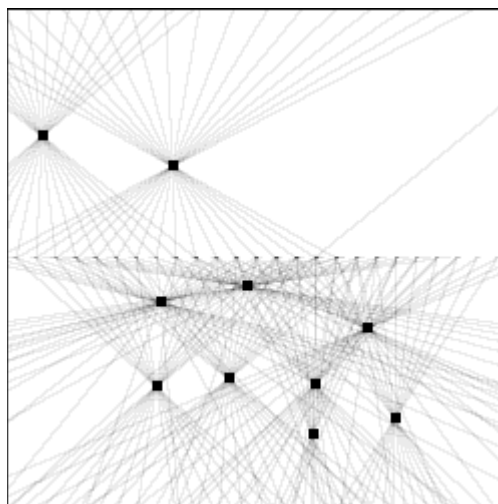


図 9.3 実験 1 の誤差

図 9.1 は実験 1 によって得られた携帯電話の密度が比較的低い場合の代表的な結果である。中央左右方向に探索体が移動している。また直線が交差した地点が携帯電話の推定位置である。図 9.2 の携帯電話実位置と重ねた図 9.3 を見ると、すべての点が誤差なく正確に推定することができていることがわかった。

表 2: 実験条件 2

探索領域	250×250
携帯電話数	100
通信半径	100
探索点の間隔	10

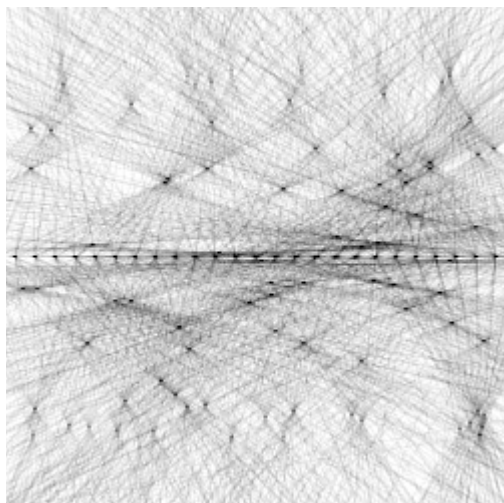


図 10.1 実験 1 の実験結果

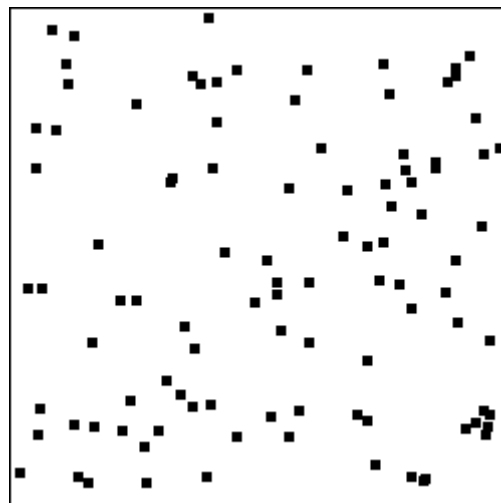


図 10.2 実験 1 の携帯電話実位置

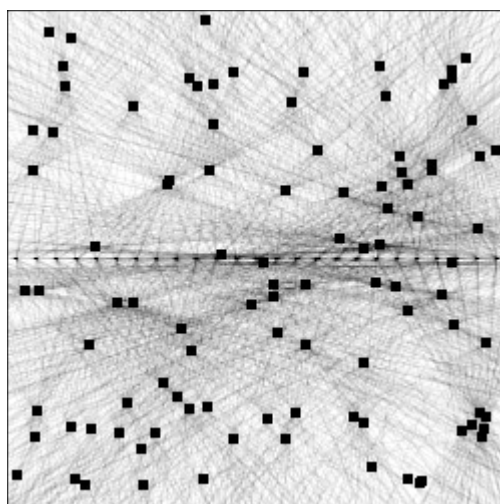


図 10.3 実験 1 の誤差

図 10.1 は携帯電話の密度が比較的高い状態での代表的な測定結果である。図 10.2 の携帯電話実位置と比較した図 10.3 を見ると、中央の探索体から上下に離れた地点の推定ができていないことがわかる。これは表 2 をみるとわかるが、携帯電話の電波が届く範囲が 100m なのに対して、探索フィールドが 250m×250m ある。すなわち中央の探索体から上下方向に±125m あり、電波を受信することができない位置が存在しているためである。

上記 2 つの実験から、受信できる範囲内の携帯電話はもれなく推定することが可能であったため、理想的な条件で尚且つ電波が届く範囲であれば携帯電話の位置推定は携帯電話の数にあまり依存しないことがわかった。

5.2 実験2の結果

表3:実験条件3

探索領域	250×250
携帯電話数	10
通信半径	100
探索角度	10°(±5°)
探索点の間隔	10

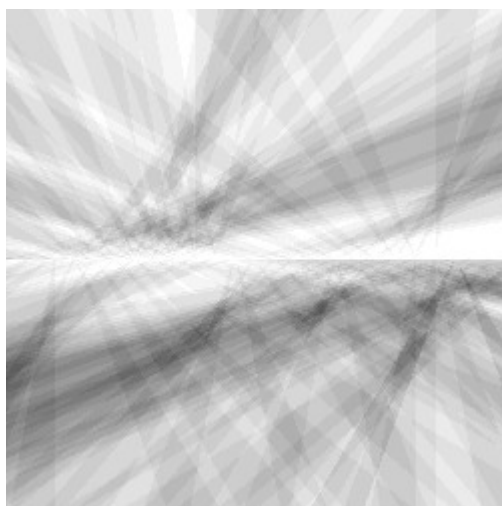


図 11.1 実験2の実験結果

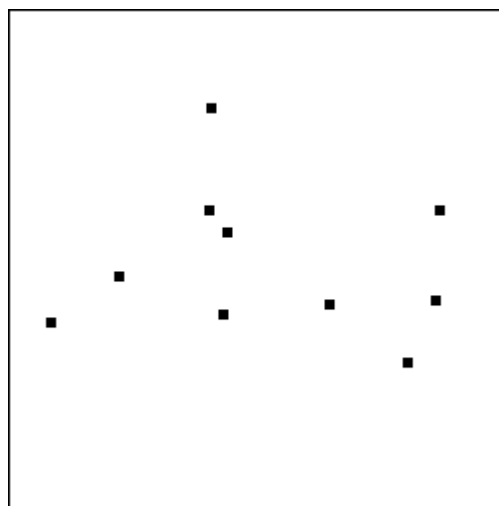


図 11.2 実験2の携帯電話実位置

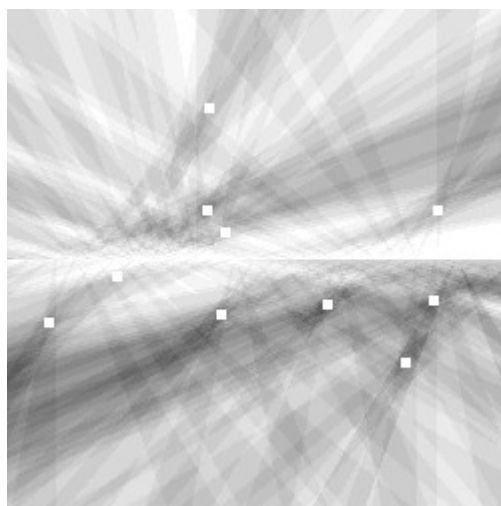


図 11.3 実験2の誤差

図 11.1 は 10 度の角度を持った探索範囲によって描かれた探索フィールドの実験結果である。携帯電話の実位置と比べた図 11.3 をみると、携帯電話の密度が低い時、携帯電話の位置特定はある程度の誤差はあるが、推定することが可能であることがわかった。

表 4: 実験条件 4

探索領域	250×250
携帯電話数	100
通信半径	100
探索角度	10°(±5°)
探索点の間隔	10

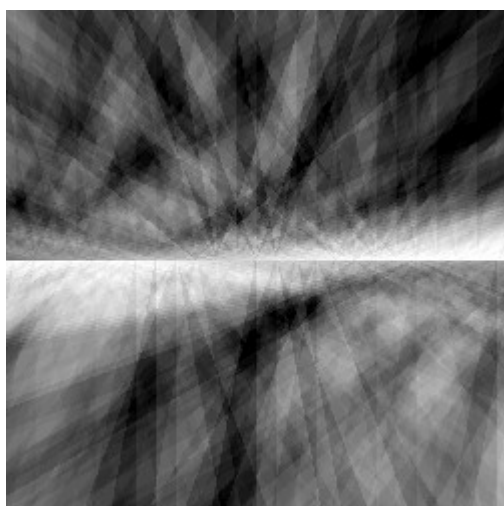


図 12.1 実験 2 の実験結果

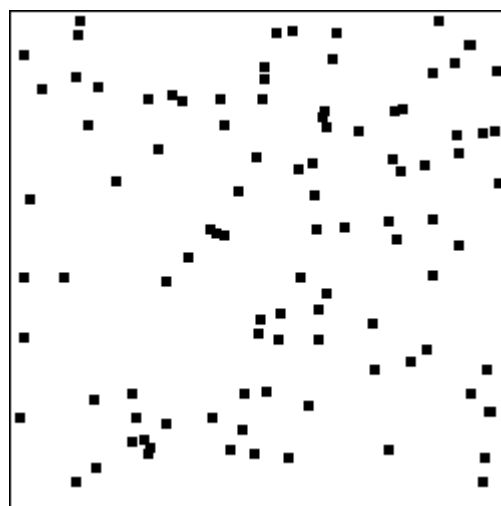


図 12.2 実験 2 の携帯電話実位置

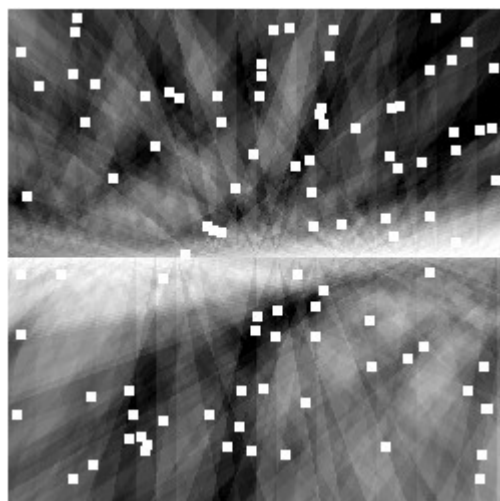


図 12.3 実験 2 の誤差

図 12.1 は携帯電話の密度が高い状態の探索結果である。図 12.1 をみてわかるように探索結果が重なり合うことで、携帯電話の正確な位置を特定することが非常に難しいことがわかる。しかし携帯電話が多く存在する範囲ではなく、携帯電話があまり存在していない地点に注目してみると、実験条件 3 の時のように大体の位置を推定することが可能であることがわかった。また周囲にまったく存在していない範囲の色はなだらかに白く、携帯がない範囲の推定に役立つことがわかった。

表 5: 実験条件 5

探索領域	250×250
携帯電話数	10
通信半径	100
探索角度	15°(±7.5°)
探索点の間隔	10

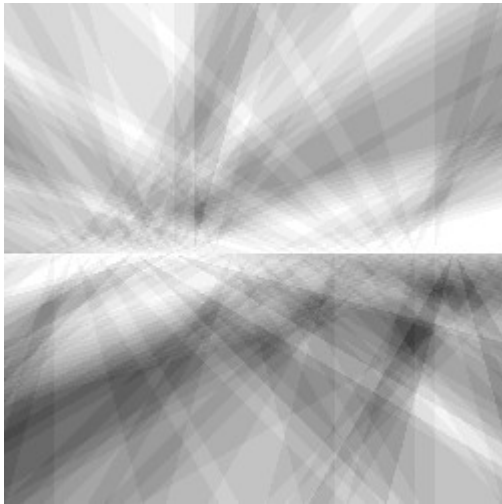


図 13.1 実験 2 の実験結果

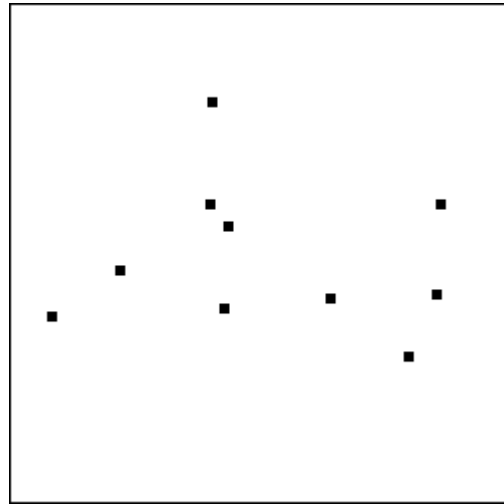


図 13.2 実験 2 の携帯電話実位置

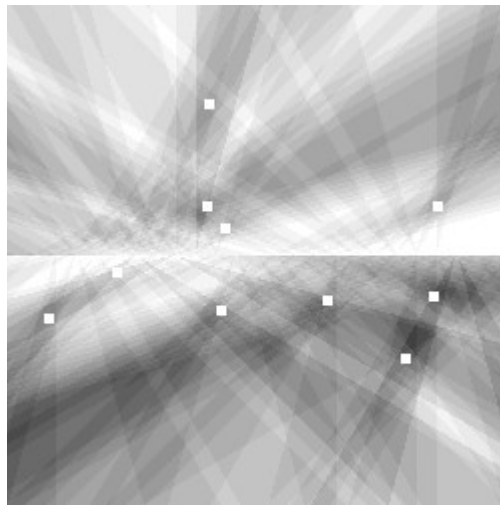


図 13.3 実験 2 の誤差

表 6: 実験条件 6

探索領域	250×250
携帯電話数	10
通信半径	100
探索角度	20°(±10°)
探索点の間隔	10

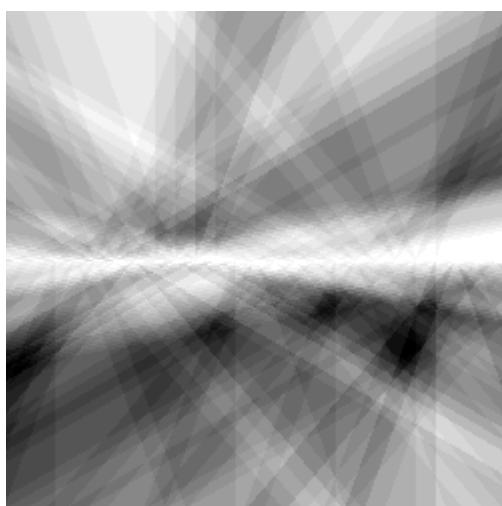


図 14.1 実験 2 の実験結果

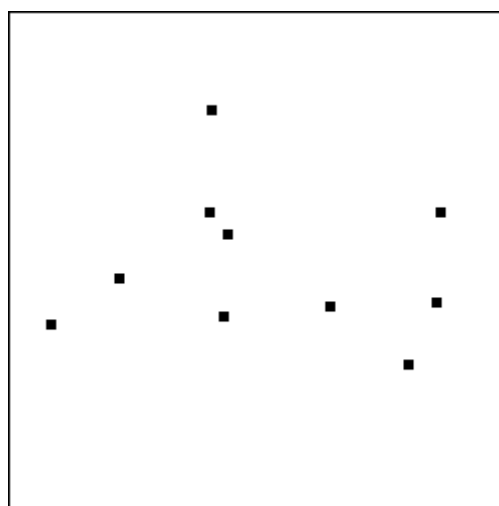


図 14.2 実験 2 の携帯電話実位置

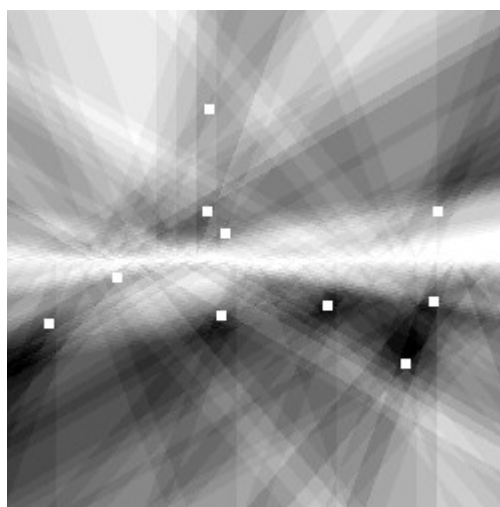


図 14.3 実験 2 の誤差

実験条件 5 と実験条件 6 は実験条件 3 の携帯電話実位置と同じものを使用し、探索角度をそれぞれ 15°、20° と増やした実験結果である。図 13.3、図 14.3 をみると探索角度が増えると推定範囲が広がっていき、携帯電話の密度が高い時と同様に場所特定が困難になっていくことがわかった。

5.3 考察

実験2によって得られた結果は予想していた以上に携帯電話の台数が増えるごとに位置推定が困難になるという結果が得られてしまった。しかし、携帯電話が少ない場合においてはある程度の精度をもった位置推定が可能であることもわかった。

この結果は人の早期発見という点において位置推定の困難さとは裏腹に役に立つものとなっている。その理由のひとつに携帯電話が多数存在している、携帯電話があまり存在しない、この二つの状態を見分けることがたやすいという事にある。人を探索する上で携帯が多く存在している範囲、すなわち人が多く取り残されている可能性がある範囲の優先度は高い。人が多く存在している可能性のある場所は必然的に人手を多くかけて探索、救助する必要がある。そのため探索する範囲内での精度が多少低くても、人が存在する間隔が狭く、探索人数も多い、比較的発見しやすい携帯電話の密集地域において精度はあまり重要でないと考えるからである。

さらに携帯電話があまり存在しない地域において位置推定がおこなえるという所が、この実験において重要なことである。人が余り存在しない場所で当てもなく搜索するというのは、人手が多く必要なことは変わらないのに、見返りとしての生存者の発見のメリットが非常に少ない。そのような範囲において場所を絞ることができることはこの手法の大きな強みである。人がもともと少ない地域の人命救助において、あまり多くの人手をかけずとも、ピンポイントで救助を行うことで生存率を大きく上げることが可能であると考えられる。

以上の点から携帯電話を探索する今回の提案手法は実際の広域災害時において、人の派遣などの割り振りや、過疎地においての効率のよい探索を行うために非常に有効な手法であると考えられる。

6 おわりに

本論文では、地上からアンテナを使い携帯電話の位置を特定することで被災者の早期発見を行う手法を提案した。実験結果より、強風などにより従来の手法が使い辛い場合、スカイメッシュを用いた緊急災害用システムを使用しなくても、探索が可能であることがわかった。1度の探索で探索することができる点も、広範囲を探索する今回の目的においては効果的であると考えられる。

問題点として、実験2において携帯が密集している場所は発見することが可能だったが密集地点から少し離れて存在している場合、ノイズのように隠されてしまうことがあった。また道などが瓦礫などによって通行が困難な箇所においては、従来の手法のほうが使い勝手がよいことや、浅い角度から瓦礫の中を探索することになる為、実際の現場では携帯電話の電波がどの程度減衰するかが、実際に実験してみないとわからない。よって今後は実際に実験を行う必要があるだろう。環境に応じて従来の手法と使い分けや併用することが必要になるだろう。

謝辞

本研究を進めるにあたり、多くの助言、指導をしてくださった三好力教授に心から御礼申し上げます。また、研究の助言や日常の議論を通じて多くの知識や示唆を頂いた三好研究室の皆様に厚く御礼申し上げます。

参考文献

- [1]警視庁 HP「東日本大震災について」<<http://www.npa.go.jp/archive/keibi/biki/index.htm>>
- [2]気象庁 HP「平成 23 年(2011 年)東北地方太平洋沖地震 ～The 2011 off the Pacific coast of Tohoku Earthquake～」<http://www.seisvol.kishou.go.jp/eq/2011_03_11_tohoku/index.html>
- [3]鈴木 裕和,兼子 陽市郎,間瀬 憲一,山崎 重光, 牧野 秀夫,「大規模災害復旧時のアドホック通信システム-スカイメッシュ-」『2006 年電子情報通信学会総合大会』
- [4]出水 宏明,三好 力,「スカイメッシュを用いた緊急災害用システム」
- [5]「CeatecJapan2012 NEC、バルーンを使った救命通信システム「緊急公助ワイヤレスソリューション」を提案 」<http://www.ceatec.com/2012/ja/news/webmagazine_detail.html?mag_vol=077>
- [6]辻 智弥、三好 力、「携帯電話を用いた位置探索手法の検討」、情報処理学会第 75 回全国大会(2013 発表予定).

付録 ソースコード

実験 1

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include <time.h>

#define XSIZE 250
#define YSIZE 250

#define TPMAX 256 //探索ポイントの最大個数
#define PPMAX 10 //携帯電話の最大個数

#define TPDIS 10 //探索ポイントの間隔

int GetRandom(int,int);
int GetRandom2(int,int,int);

int main(void){
    int map[XSIZE][YSIZE]={0};
    //探索地点の X,Y 座標
    double TP[TPMAX][2];
    //携帯電話の X,Y 座標、電波の距離
    double PP[PPMAX][3];
    int x1,x2,xs,xl,m,n;

    srand((unsigned int)time(NULL));

    FILE *fp,*fp2,*fp3;
    fp = fopen("data.txt","w");
    fp2= fopen("MAPDATA.csv","w");
    fp3= fopen("MAPDATA2.csv","w");

    int i,j,k,l;

    //探索ポイントの決定
    for(i=0;i<YSIZE/TPDIS*2;i++){
        TP[i][0]=XSIZE/2;
        TP[i][1]=i*TPDIS-YSIZE/2;
    }
    //携帯電話の位置決定
    for(i=0;i<PPMAX;i++){
        PP[i][0]=GetRandom2(0+5,XSIZE-1-5,XSIZE/2);
        PP[i][1]=GetRandom(0+5,YSIZE-5);
        PP[i][2]=GetRandom(100,100);
    }

    //map 計算
    for(i=0;i<YSIZE/TPDIS*2;i++){
        for(j=0;j<PPMAX;j++){
            if((TP[i][0]-PP[j][0])*
                (TP[i][0]-PP[j][0])+(TP[i][1]-PP[j][1])*
                (TP[i][1]-PP[j][1])<=PP[j][2]*PP[j][2]){
                for(k=0;k<YSIZE;k++){
                    x1=(int)((double)k- TP[i][1])*
                        (TP[i][0]-PP[j][0])/(TP[i][1]-PP[j][1])+TP[i][0];
                    x2=(int)((double)k+1.0-TP[i][1])*
                        (TP[i][0]-PP[j][0])/(TP[i][1]-PP[j][1])+TP[i][0];
                    if (x1<=x2){xs=x1;xl=x2;}
                    else {xs=x2;xl=x1;}
                    if (x1<=x2 && TP[i][0]>x1) xs=TP[i][0];
                    if (x1<=x2 && XSIZE<=x1) xl=XSIZE-2;
                    if (x1>x2 && TP[i][0]>x2) xs=TP[i][0];
                    if (x1>x2 && XSIZE<=x2) xl=XSIZE-2;
                    for(l=xs;l<=xl && l<XSIZE;l++){
                        map[l][k]++;
                    }
                }
            }
            else {
            }
        }
    }
    for(i=0;i<PPMAX;i++){
        fprintf(fp,"%3d,%3d,%3d\n",(int)PP[i][0],(int)PP[i][1],
            (int)PP[i][2]);
    }
    for(i=0;i<XSIZE;i++){
        for(j=0;j<YSIZE;j++){
            fprintf(fp2,"%d,",(int)map[i][j]);
        }
        fprintf(fp2,"\n");
    }
    for(i=0;i<XSIZE;i++){
        for(j=0;j<YSIZE;j++){
            map[i][j]=0;
        }
    }
    for(i=0;i<XSIZE;i++){
        for(j=0;j<YSIZE;j++){
            for(k=0;k<PPMAX;k++){
                if((int)PP[k][0]==i && (int)PP[k][1]==j){
                    for(m=i-2;m<=i+2;m++){
                        for(n=j-2;n<=j+2;n++){
                            map[m][n]=255;
                        }
                    }
                }
            }
        }
    }
    for(i=0;i<XSIZE;i++){
        for(j=0;j<YSIZE;j++){
            fprintf(fp3,"%d,",(int)map[i][j]);
        }
        fprintf(fp3,"\n");
    }
    fclose(fp);
    fclose(fp2);
    fclose(fp3);
    return 0;
}

int GetRandom(int min,int max)
{
    return min + (int)(rand()*((max-min+1.0)/(1.0+RAND_MAX)));
}
```

```

int GetRandom2(int min,int max,int c)
{
    int r;
    while(1){
        r=min + (int)(rand()*(max-min+1.0)/(1.0+RAND_MAX));
        if(c!=r) break;
    }
    return r;
}

```

実験 2

```

#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include <time.h>

#define XSIZE 251
#define YSIZE 250

#define TPMAX 256 //探索ポイントの最大個数
#define PPMAX 10 //携帯電話の最大個数

#define TPDIS 10 //探索ポイントの間隔
#define CANG 10 //探索の角度

int GetRandom(int,int);
int GetRandom2(int,int,int);

int main(void){

    int map[XSIZE][YSIZE]={0};
    int map0[XSIZE][YSIZE]={0};
    //探索地点の X,Y 座標
    double TP[TPMAX][2];
    //携帯電話の X,Y 座標、電波の距離
    double PP[PPMAX][3];

    double x1,x2;
    int m,n;
    double rad,a1,a2;

    srand((unsigned int)time(NULL));

    FILE *fp,*fp2,*fp3;
    fp = fopen("data.txt","w");
    fp2= fopen("MAPDATA.csv","w");
    fp3= fopen("MAPDATA2.csv","w");

    int i,j,k,l;

    //探索ポイントの決定
    for(i=0;i<YSIZE/TPDIS*2+1;i++){
        TP[i][0]=XSIZE/2;
        TP[i][1]=i*TPDIS-YSIZE/2;
    }
    //携帯電話の位置決定
    for(i=0;i<PPMAX;i++){
        PP[i][0]=GetRandom2(0+5,XSIZE-1-5,XSIZE/2);
        PP[i][1]=GetRandom(0+5,YSIZE-5);
        PP[i][2]=GetRandom(100,100);
    }

    //map 計算
    for(i=0;i<YSIZE/TPDIS*2;i++){
        for(k=0;k<360;k=k+CANG){
            rad=(double)(k%90)*3.141592/180.0;
            x1=tan((double)rad);
            rad=(double)((k+CANG)%90)*3.141592/180.0;
            x2=tan(rad);

```

```

        for(j=0;j<PPMAX;j++){
            //距離判定
            if((TP[i][0]-PP[j][0])*
                (TP[i][0]-PP[j][0])+(TP[i][1]-PP[j][1])*
                (TP[i][1]-PP[j][1])<=PP[j][2]*PP[j][2]){
                //角度判定
                if(k<90 && k+CANG<90){
                    if((PP[j][0]-TP[i][0])*x1+TP[i][1]<=PP[j][1] &&
                        (PP[j][0]-TP[i][0])*x2+TP[i][1]>PP[j][1]){
                        for(m=TP[i][0];m<XSIZE;m++){
                            a1=(m-TP[i][0])*x1+TP[i][1];
                            a2=(m-TP[i][0])*x2+TP[i][1];
                            if(a2>=YSIZE) a2=YSIZE-1;
                            for(n=(int)a1;n<(int)a2;n++){
                                map[m][n]++;
                            }
                        }
                    }
                }
            }
            else if(k>=90 && k<180 && k+CANG<180){
                if((PP[j][1]-TP[i][1])*x1+TP[i][0]>PP[j][0] &&
                    (PP[j][1]-TP[i][1])*x2+TP[i][0]<=PP[j][0]){
                    for(m=TP[i][1];m<YSIZE;m++){
                        a1=(m-TP[i][1])*x1*-1+TP[i][0];
                        a2=(m-TP[i][1])*x2*-1+TP[i][0];
                        if(a2<=0) a2=0;
                        for(n=(int)a2;n<(int)a1;n++){
                            map[n][m]++;
                        }
                    }
                }
            }
            else if(k>=180 && k<270 && k+CANG<270){
                if((PP[j][0]-TP[i][0])*x1+TP[i][1]>PP[j][1] &&
                    (PP[j][0]-TP[i][0])*x2+TP[i][1]<=PP[j][1]){
                    for(m=TP[i][0];m>=0;m--){
                        a1=(m-TP[i][0])*x1+TP[i][1];
                        a2=(m-TP[i][0])*x2+TP[i][1];
                        if(a2<=0) a2=0;
                        for(n=(int)a2;n<(int)a1;n++){
                            map[m][n]++;
                        }
                    }
                }
            }
            else if(k>=270 && k<360 && k+CANG>=270 &&
                k+CANG<360){
                if((PP[j][1]-TP[i][1])*x1+TP[i][0]<=PP[j][0] &&
                    (PP[j][1]-TP[i][1])*x2+TP[i][0]>PP[j][0]){
                    for(m=TP[i][1];m>=0;m--){
                        a1=(m-TP[i][1])*x1*-1+TP[i][0];
                        a2=(m-TP[i][1])*x2*-1+TP[i][0];
                        if(a2>=XSIZE) a2=XSIZE-1;
                        for(n=(int)a1;n<(int)a2;n++){
                            map[n][m]++;
                        }
                    }
                }
            }
        }
    }
    //携帯の座標を出力
    for(i=0;i<PPMAX;i++){
        fprintf(fp,"%3d,%3d,%3d\n",(int)PP[i][0],(int)PP[i][1],
            (int)PP[i][2]);
    }
    //MAPを出力
    for(i=0;i<XSIZE;i++){
        for(j=0;j<YSIZE;j++){
            fprintf(fp2,"%d,",(int)map[i][j]);
        }
        fprintf(fp2,"\n");
    }
}

```

```

for(i=0;i<XSIZE;i++){
    for(j=0;j<YSIZE;j++){
        map[i][j]=0;
    }
}

for(i=0;i<XSIZE;i++){
    for(j=0;j<YSIZE;j++){
        for(k=0;k<PPMAX;k++){
            if((int)PP[k][0]==i && (int)PP[k][1]==j){
                for(m=i-2;m<=i+2;m++){
                    for(n=j-2;n<=j+2;n++){
                        map[m][n]=255;
                    }
                }
            }
        }
    }
}

for(i=0;i<XSIZE;i++){
    for(j=0;j<YSIZE;j++){
        fprintf(fp3,"%d,",(int)map[i][j]);
    }
    fprintf(fp3,"\n");
}

fclose(fp);
fclose(fp2);
fclose(fp3);

return 0;
}

```

```

int GetRandom(int min,int max)
{
    return min + (int)(rand()*(max-min+1.0)/(1.0+RAND_MAX));
}

int GetRandom2(int min,int max,int c)
{
    int r;
    while(1){
        r=min + (int)(rand()*(max-min+1.0)/
(1.0+RAND_MAX));
        if(c!=r) break;
    }
    return r;
}

```