

平成 25 年度 特別研究報告書

携帯端末を利用した電波探索手法

龍谷大学 理工学部 情報メディア学科

T100408 辻 拓也

指導教員 三好 力 教授

内容概要

日本は昔から地震大国であり、いつ東日本大震災のような巨大地震が日本のどんな場所で起こるか予想できない。建物が崩れるような大きな地震が起こり、人がその下敷きになってしまった場合、72時間を過ぎると極端に生存率が下がってしまうため72時間以内に発見することが非常に重要である。そこで携帯電話が発する電波を手がかりに下敷きになった人を検索する手法を提案する。現代ではほとんどの人が携帯電話を常備しているため電波の発信源の近くに人間がいることが想定される。電波の発信源を特定できれば災害時の救出が効率よく行えるのではないかと考えた。

目次

第1章	はじめに	
1.1	本研究の背景	1
1.2	研究目的	1
第2章	既存技術	2
2.1	緊急公助ワイヤレスソリューション(NEC)	2
2.2	スカイメッシュを利用した位置探索手法	3
2.3	指向性アンテナを使った位置探索手法	4
2.4	問題点と改善策	5
第3章	利用する技術の概要	6
3.1	電波強度	6
3.2	最小二乗法	7
第4章	提案手法	8
4.1	最小二乗法による電波の発信源特定方法	8
4.1.1	円の作成	9
4.1.2	円の中心の推定	10
第5章	実験	11
5.1	シミュレーションでの提案手法の確認	11
5.1.1	実験目的	11
5.1.2	実験手順	11
5.1.3	実験結果	11
5.2	屋外での電波発信源の特定	13
5.2.1	実験目的	13
5.2.2	開発環境	13
5.2.3	実験手順	13
5.2.4	「屋外での電波発信源の特定」の実験結果	13
5.2.5	実験目的	15
5.2.6	実験手順	15
5.2.7	「屋外での電波発信源の特定」の実験結果	15
5.2.8	考察	16
5.3	シミュレーション実験	17
5.3.1	実験目的	17
5.3.2	実験手順	17
5.3.3	実験結果	19
第6章	まとめ	20
	謝辞	21
	参考文献	22
	付録	

第1章はじめに

1.1 本研究の背景

日本は昔から地震が多く発生する場所である。近年で言えば2011年3月11日に発生した東日本大震災が記憶に新しい。この地震により大きな津波が発生し多くの家や人が被害を受けた。消防庁より発表されている災害情報詳報によると平成25年7月2日時点での人的被害は死者: 16019, 行方不明: 3805, 重傷: 666, 軽傷: 5190, 重軽傷: 612、建物被害は全壊: 118621, 半壊: 181801, 一部破損: 621013という甚大な被害をもたらした。また東日本大震災以後毎月の地震発生回数が増えている。

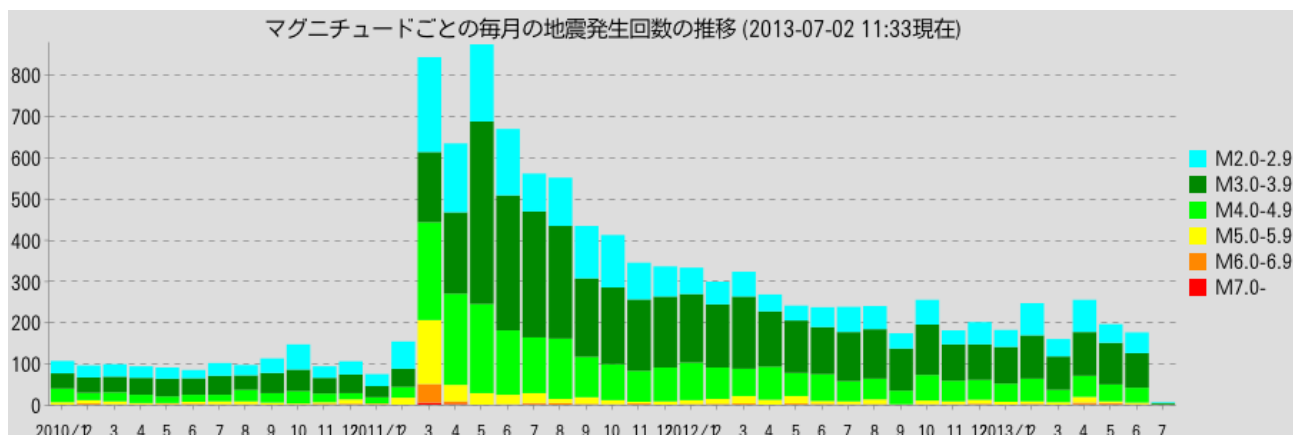


図1 毎月の地震発生回数の推移

いつ東日本大震災のような地震が日本のどんな場所で起こるかは予想できない。そのため地震に備えた対策が必要となっている。建物などにも崩れたりしないような対策がこれから施されていくはずだが、100%崩れないわけではない。大きな地震が起こり、建物などが倒壊した場合、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原因で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が起きた場合、早期に人を発見することが重要である。

1.2 研究目的

1.1節でも記述したように、日本は地震大国であり、いつ地震が起きるかわからない環境の中で私たちは暮らしている。地震などの災害が起こり建物などが倒壊した場合、多くの人々が建物の下敷きになることが予想されるが、そうなった場合72時間以内に救出する必要がある。

現代ではほとんどの人間が携帯電話を常備しているため電波の発信源の近くに人間がいることが想定される。これを災害時に建物が倒壊した場合などに行うことで、瓦礫に埋もれた人々の位置を迅速に特定できると考える。瓦礫に埋もれた人々の位置を迅速に特定することで、搜索範囲を小さくすることができ、効率良く人命救助することができる。更に搜索時間を減らすことで、余った時間を別の範囲の搜索に使うことができ、より多くの人々を救助することができる。

第2章 既存技術

2.1 緊急公助ワイヤレスソリューション(NEC)

震災発生直後に、倒壊した建物の上空に通信アンテナを搭載したバルーンやクレーンを上げることで、そのアンテナ配下の携帯電話を探し出し、範囲内の被災者が持つ携帯電話から発信されている信号を受信することで携帯電話の在りかが分かる。



図2 緊急公助ワイヤレスソリューションの例

バルーン側では携帯電話の通話状況をキャッチすることができる。例えば、「発信中」や「メール送信中」などの携帯端末が何かを発信している通信状況も把握することができる。それを把握することで、生存していることを確認でき、且つ助けを求めている人をいち早く見つけることができる。

2.2 スカイメッシュを利用した位置探索手法

大規模災害発生時に基地局や電波などに障害が発生することにより、深刻な災害発生直後からある程度復旧するまでの数週間において、通信インフラなどを使用することができない場合がある。実際に2004年10月に発生した新潟県中越地震では、光ファイバーケーブルの切断などが起こり通信インフラの断絶が起こった。そして被害状況の把握に時間を要したり、被災地外からアクセスしにくい状況が続き住民の安否確認に時間が掛かるといったことも起きる。これらのことから被災者のための様々な情報連絡、復旧活動のための通信網、通信手段が素早く簡易に確立されることが必要となってくる。

そこで登場するのが「気球を用いたアドホック通信システムスカイメッシュ」である。スカイメッシュでは通信機器を気球に吊るし、地上高50m～100mに気球を保留することにより、簡易基地局を空に打ち上げることで通信環境を構築する。

災害時において救助活動などの目的で用いられる。



図3 スカイメッシュ用気球

スカイメッシュによって通信環境を構築する際に、各ノードの受信範囲をわざと少し重ねる。それによって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事ができる。そして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もできる。

2.3 指向性アンテナを使った位置探索手法

車の上に強い指向性を持ったアンテナを乗せ、360度回転させながら走らせることによって、移動しながら携帯電話の位置を特定するという手法である。指向性アンテナだけでは携帯電話の位置が把握できても自分の向いている方角がわからない。そこで車の上に指向性アンテナを乗せ走らせることによって自分の方角が把握できるというものである。

以下に携帯電話の数が少ない場合の実験結果を示す。

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4 実験条件[引用1]

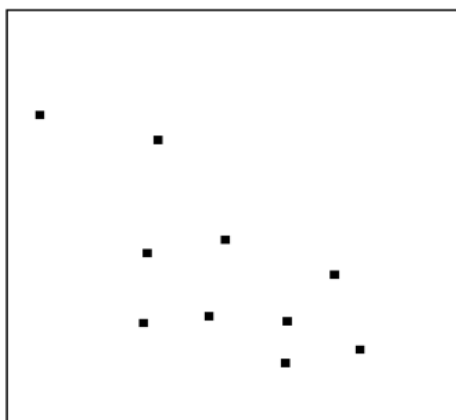


図5 携帯電話実位置[引用2]

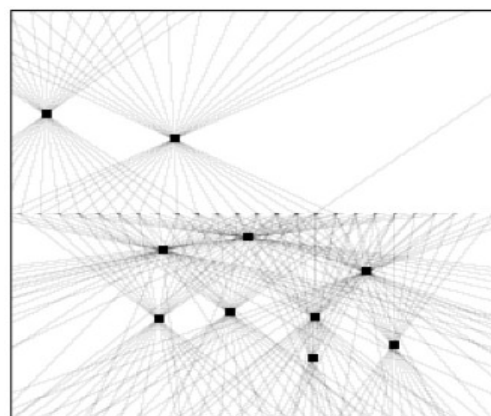


図6 実験結果[引用3]

携帯電話の数が少ない場合や携帯電話の数が多の場合などの実験を繰り返し行なった結果、この既存技術は電波の発信源の数が少ないときには有効に使えるということがわかった。携帯電話の位置が分かればそこを重点的に搜索し、人手や搜索時間も少なくすむ上に被災者が見つかる可能性が高い。

では、電波の発信源の数が多い時には使えないのかというとそうではなく、電波の発信源の数が多い場合は電波の発信源の位置を正確に把握することはできないが、おおよその位置は分かる。なぜなら電波の発信源が存在する、存在しないという二つの状態を見分けることができるからである。電波の発信源が存在しない場所は人がいる可能性は低く、電波の発信源が存在する場所は人がいる可能性が高い。その場所を搜索すれば人がいる可能性が高く、搜索時間も減らせる。

2.4 問題点と改善策

2.1 節の場合一番の問題が風速である。風速が8mほどまでならバルーンは安定するがそれ以上の風速のとき動作が不安定になるなど、その日の気象状況に大きく左右されることがある。また、バルーンを空に打ち上げるためには大量のヘリウムガスを必要とする。しかし、災害が起こったときに大量のヘリウムガスを用意するのは難しい。バルーンの代わりにクレーンなどを使用する場合でも、震災が起こった場所までクレーンを素早く移動させることは難しい上に瓦礫などが散乱している状態の場所にクレーンを用意することも難しい。

2.2 節においても、2.1 節での問題点と同様に、気球を空に打ち上げる時に風速が強すぎる場合は打ち上げることが不可能となってしまふ。

2.3 節の場合自動車を使うため、地震などの災害が起きた場合、建物の倒壊などにより自動車で現地を走れないということが起きる。指向性アンテナを人が担げばいいのではないかというアイデアもあるかもしれないが指向性アンテナは重量があり、なかなか人が担げるものでないし、仮に指向性アンテナを担いで歩いたとしても車のような速さが出ないため、自分がどの方角を向いているかが把握できない。このような問題を解決するために、GPS 機能と電波強度を測れる機能を持った小型で安価なタブレットを用いて電波の発信源を特定する事を考えた。これならば地震などの災害時に建物が倒壊して瓦礫が散乱していても、人が手に持ち歩くだけなので問題ないし、人が手に持つのが危険な場所の場合、GPS 機能と電波強度と測れる機能を持たせたロボットなどに代用してもいい。GPS 機能と電波強度を測れる機能を用いて電波の発信源を特定できるのか。そして特定できたときの誤差などを調べた。

第3章 利用する技術の概要

3.1 電波強度

信号レベルというのは電波の強度のことで単位は dB(デジベル)。

dB(デジベル)はある基準に対して、どれくらい大ききなのかを対数で表したもので、単位がマイナスの値ならある基準より小さいということである。デジベルは目的に応じて基準を設けて、それぞれの表示方法が決められている。

dBf

1fW(1フェムトワット)を0dBとして、電力の大きさを dB で表したもの。無線通信の分野で受信機の感度を表現する場合に、dBm では数値が小さくなりすぎるためこの単位が使われる。

dBm

1mW を0dBとして、電力の大きさを dB で表したもの。例えば、600 オームの抵抗負荷に1mWの電力を供給するのに必要な交流電圧は、約0.775Vrmsである。

など他にも多く存在する。

dB(デジベル)は通信の世界ではレベルの単位としてよく使われている。

上記に書かれている dBm という単位の m は 1mV 基準の m という意味である。

ここに電力を表す計算式を示す。

電力を表す計算式

$dBm = 10 \log(\text{電力} / \text{基準電力})$ 基準電力: 1 mV log の根は 10

電力が 1mV の場合 0dBm となります。

わかりやすく表に表すと

表1 デジベルと電力の関係

dBm	電力
-30dBm	0.001mW
-20dBm	0.01mW
-10dBm	0.1mV
0dBm	1mW
+10dBm	10mW
+20dBm	100mW
+30dBm	1W
+40dBm	10W
+50dBm	100W
+60dBm	1KW

3.2 最小二乗法

最小二乗法とは別名、最小自乗法ともいうが「個々のデータと論理的な中心線の差を二乗したものの合計が最小になるような式がもっとも相関度の高い(もっともらしい)数式である」という考え方のもとに行われる式である。図を交えて説明すると本来直線上になると思われたデータが誤差のためにばらつきがあったとする。図7ではデータを赤い点で示している。このデータをD1,D2,D3,D4とする。誤差のためにばらつきがあるが、そのデータに対して $y = ax + b$ を求めたい。どうやってそのような直線を求めるのかというと求めたい直線を仮に a とおく。そしてその仮の直線 a と D1,D2,D3,D4 の差(距離)を求める。今回この差(距離) $d1, d2, d3, d4$ を青い線で表している。このとき $S = d1^2 + d2^2 + d3^2 + d4^2$ とする。S の値を最小にするような直線 $y = ax + b$ を正しい直線とするというのが最小二乗法である。

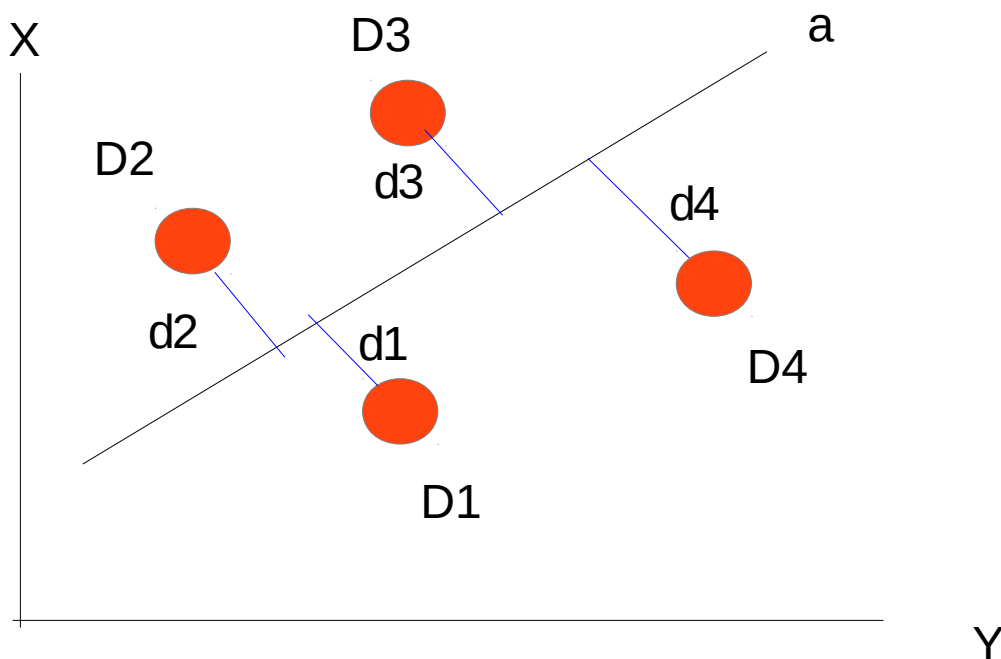


図7 最小二乗法

第4章 提案手法

4.1 最小二乗法による電波の発信源特定方法

GPS 機能と電波強度の測定ができる機能(今回はタブレットを使用)を用意した。提案手法の流れは以下の通りである。

- ①多数の電波の信号レベルと現在位置を測定できる測定点(タブレット)をランダムに配置する。
 - ②測定点ひとつひとつで、その場所の緯度と経度を GPS で取得し電波の信号レベル(単位は dBm)とともに記録する。
 - ③それぞれの測定点で取得した電波の信号レベルの値が近い測定点を選択する。
 - ④選択した測定点は電波の信号レベルと同時に緯度と経度も取得しているなので、その緯度と経度の情報を用いて最小二乗法を行う。
 - ⑤最小二乗法を行い、選択したどの測定点からも一番距離が小さくなる円を作る。
 - ⑥円の中心を求める。
- 以下で詳細を説明する。

4.1.1 測定点による電波の信号レベルの取得

電波の発信源の周りに多数の測定点を用意する。図8の場合、赤い点が電波の発信源、周りの黒い点が測定点、円が電波の範囲としている。測定点一つ一つが電波の信号レベル(dBm)を取得する。

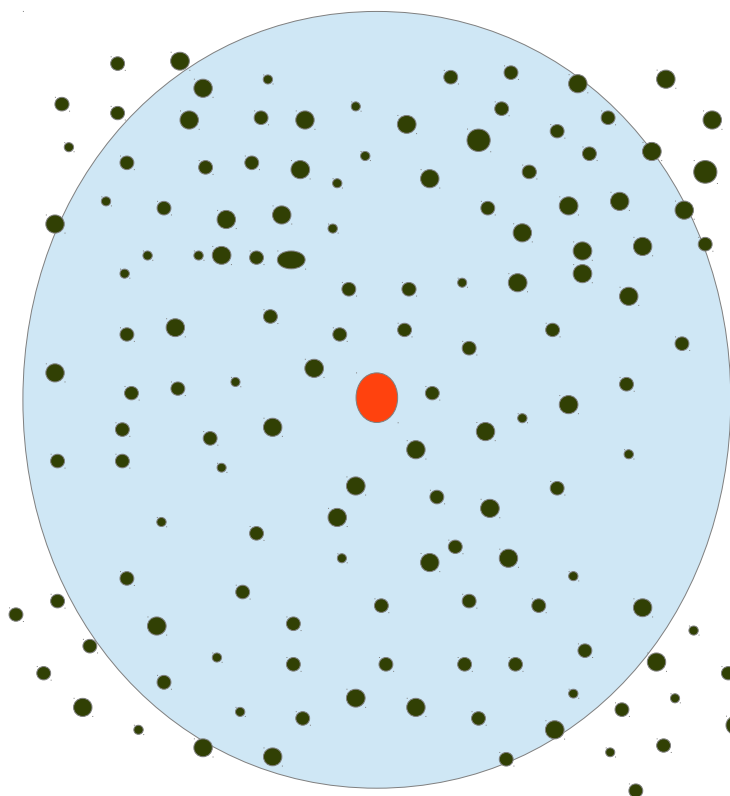


図8 実験時のイメージ図

4.1.2 円の作成

測定点がひとつひとつ電波の信号レベルを取得すれば、その信号レベルの数值が近いもの同士で最小二乗法を行い、選択したどの測定点からも距離が一番短い円を作る。図9でいうと、電波強度が10dBm、11dBm、12dBm、10dBm、11dBmの測定点を選択し、この5つの測定点で最小二乗法を行いどの測定点からも距離が一番短い円を作る。この5つの測定点を使用したのは電波の信号レベルが非常に近かったからであり、10dBmと36dBmといった電波の信号レベルが離れているもの同士で最小二乗法を行わない。

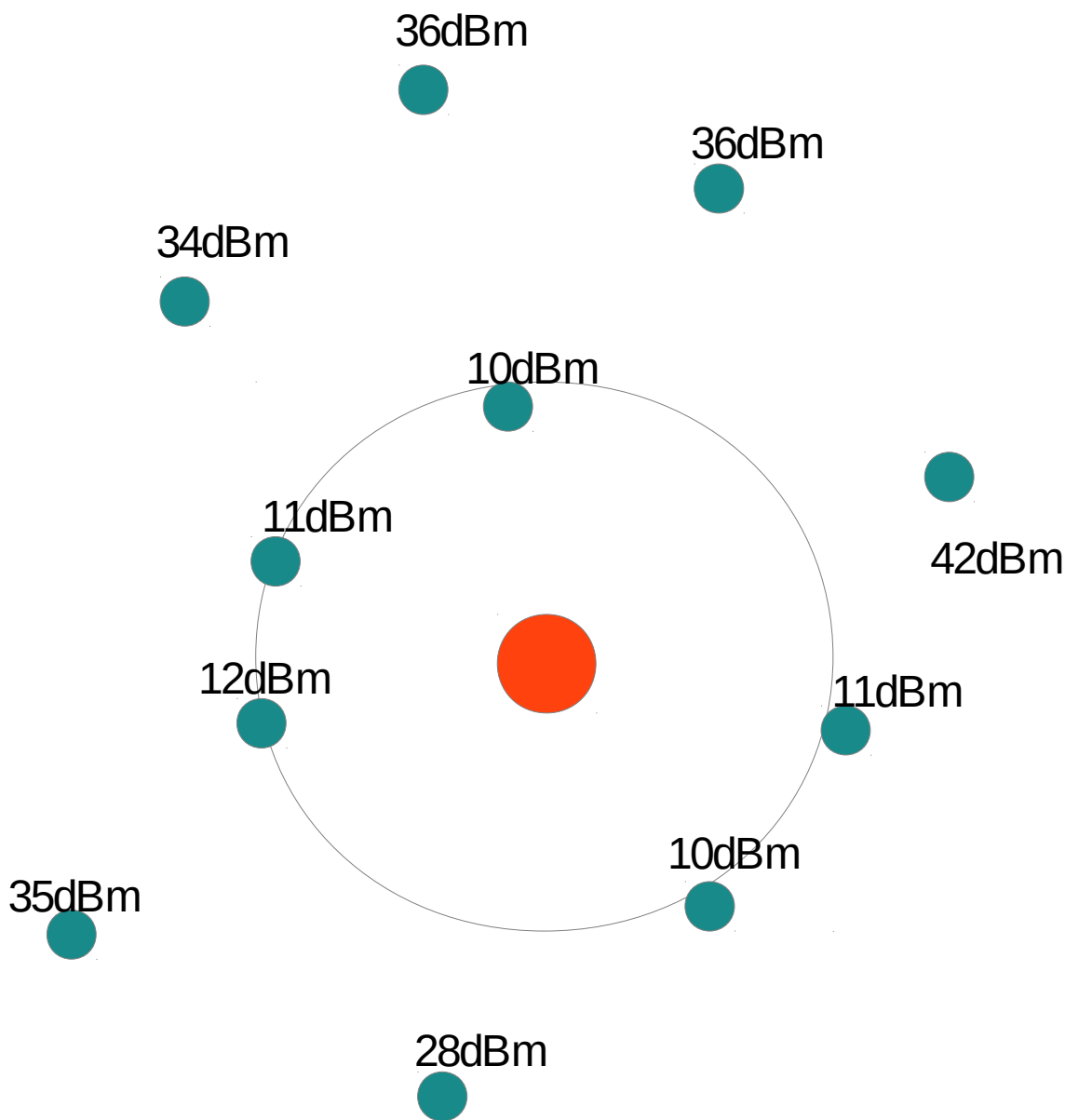


図9 円の作成時のイメージ図

4.1.3 円の中心の推定

円に AB、CD という二本の線を引きそれぞれの線の垂直二等分線を引く。
その垂直二等分線の交わったところが円の中心である。

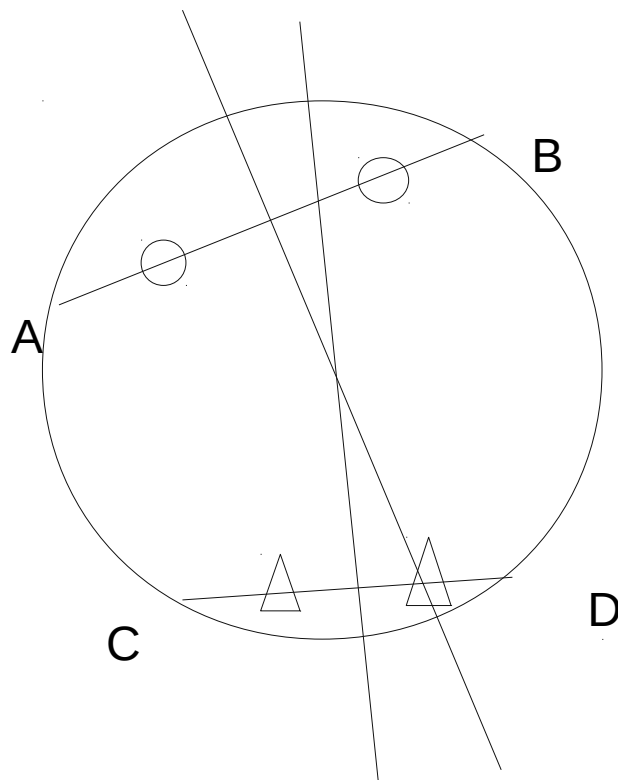


図 10 円の中心

第5章 実験

5.1 シミュレーションでの提案手法の確認

5.1.1 実験目的

提案手法で発信源が特定できる事、また電波強度の値の幅をどの程度にすれば正確に円の中心が特定できるのかについて確認するため、この実験を行った。

5.1.2 実験手順

範囲を x 座標を 0 から 40、y 座標 0 から 40 にし、(20,20)に置いた電波の発信源の電波強度を-20dBm とする。電波強度を-20dBm にしたのは屋外の実験で使用したアクセスポイントの電波強度に合わせるためである。範囲の中でランダムに配置した 30 箇所電波強度を測定した。その値から電波強度の差が約 5dBm、約 10dBm、約 30dBm の測定点だけを 11 箇所選択し、最小二乗法で中心を求めるという実験を 3 回行った。11 箇所選択するというのは、予備実験より 11 箇所より少ない測定点のデータで計算すると精度が落ちるからである。

5.1.3 実験結果

表 2 電波強度の差が約 5dBm

X	Y	dBm
12.65	18.2	-92.7
13.82	16.19	-92.59
25.1	26.01	-92.8
28.04	20.04	-92.85
20.09	27.12	-92.54
16.55	21.43	-88.69
27.8	27.08	-93.33
16.55	26.17	-92.51
19.65	27.98	-92.84
26.11	15.68	-92.67
20.59	12.13	-92.81

表 3 電波強度の差が約 10dBm

X	Y	dBm
21.56	21.47	-77.89
17.79	18.53	-83.49
16.62	18.7	-88.35
18.65	22.65	-85.63
17.7	23.36	-89.53
18.4	16.35	-89.34
17.64	18.7	-83.8
23.67	17.82	-89.93
20.28	17.6	-81.32
16.07	18.15	-90.07
20.33	17.84	-78.5

表 4 電波強度の差が約 30dBm

X	Y	dBm
19.3	18.02	-77.22
13.85	21.67	-92.17
19.33	22.31	-81.2
21.35	23.8	-89.44
19.08	18.81	-61.29
19.01	17.4	-84.43
17.97	20.21	-76.23
21.1	18.4	-74.37
19.54	17.92	-77.69
22.57	19.94	-82.8
7.65	32.6	-93.76

以下に計算で求めた推定中心を示す。今回の実験では中心を(20,20)に設定している。

表 5 電波強度の差が約 5dBm

x 座標	20.31
y 座標	19.88

表 6 電波強度の差が約 10dBm

x 座標	19.75
y 座標	19.87

表 7 電波強度の差が約 30dBm

x 座標	15.01
y 座標	25.85

シミュレーションで行った実験だが、電波強度の差を 5dBm、10dBm の範囲に留めた測定点の x 座標、y 座標だけを選択し計算した結果、(20.31,19.88),(19.75,19.87)となり中心(20,20)からずれることなく求めることができた。しかし、電波強度の差が 30dBm の範囲の測定点を選択し計算したところ(15.01,25.85)となり電波強度の差が 5dBm、10dBm の範囲で計算して求めた結果に比べ中心(20,20)からずれてしまった。このことから電波強度が 30dBm 離れている測定点の値を選択すると計算で求めた中心の値は、ずれてしまうことがわかる。この提案手法で中心を求める場合、どんなに電波強度が離れていても 10dBm 以内の測定点を選択して計算する必要があると推測する。

5.2 屋外での電波発信源の特定

5.2.1 実験目的

実際に屋外で電波強度や GPS から得る位置情報などを用いて、電波の発信源が正確に特定できるのかを調べた。

5.2.2 開発環境

今回電波強度や自分の位置情報を取得するアンドロイドアプリを自作し、それを使用した。その開発環境を以下に示す。

OS	Ubuntu12.10
開発ツール	eclipse
開発言語	Java
タブレットの Android バージョン	3.2

5.2.3 実験手順

実験場所は龍谷大学のテニスコート近くの建物で行った。龍谷大学のテニスコートがある矢印の場所を今回の電波の発信源の場所とし、ランダムにアンドロイド端末で電波の強度と自分の位置情報を取得した。ランダムに 30 箇所測定し、電波強度の値が -75dBm \sim -70dBm という電波強度の値の差が 5dBm の測定点だけを選択し、提案手法により中心を求めた。電波強度が -75dBm \sim -70dBm にした理由はシミュレーション実験結果より、電波強度の差が 10dBm 以内なら問題なく円の中心を特定できると予想した。図 11 に真の発信源を地図上にマッピングした図を示す。

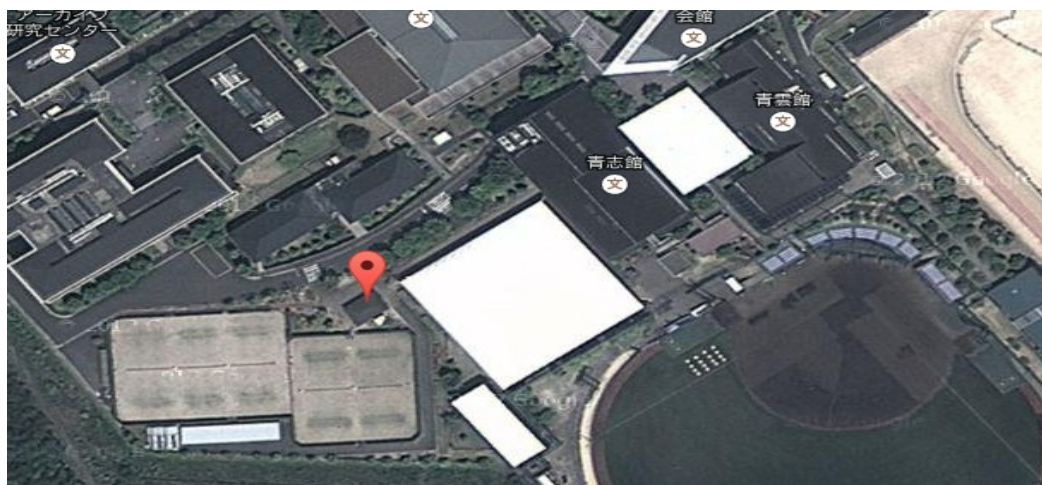


図 11 真の発信源

5.2.4 「屋外での電波発信源の特定」の実験結果

測定結果の内 -75dBm \sim -70dBm の 11 箇所を表 2 に示す。

表 8 測定結果 1

緯度	経度	dBm
34.9625333549	135.93978188	-74
34.9624962249	135.94012116	-73
34.9625025425	135.94008758	-74
34.962430475	135.93966979	-75
34.9624101472	135.93942902	-70
34.9625853404	135.94022225	-71
34.9623713367	135.93900027	-72
34.9627705075	135.93946633	-73
34.9608992228	135.94080905	-72
34.9623391077	135.93956577	-71
34.9626251365	135.93993852	-71

表 1 のデータから提案手法を用いて計算で求めた推定中心を表 9 に示す。

表 9 推定中心 1

緯度	34.964036942
経度	135.9385767

推定中心をグーグルマップを用いて地図にマッピングした結果を図 12 に示す。

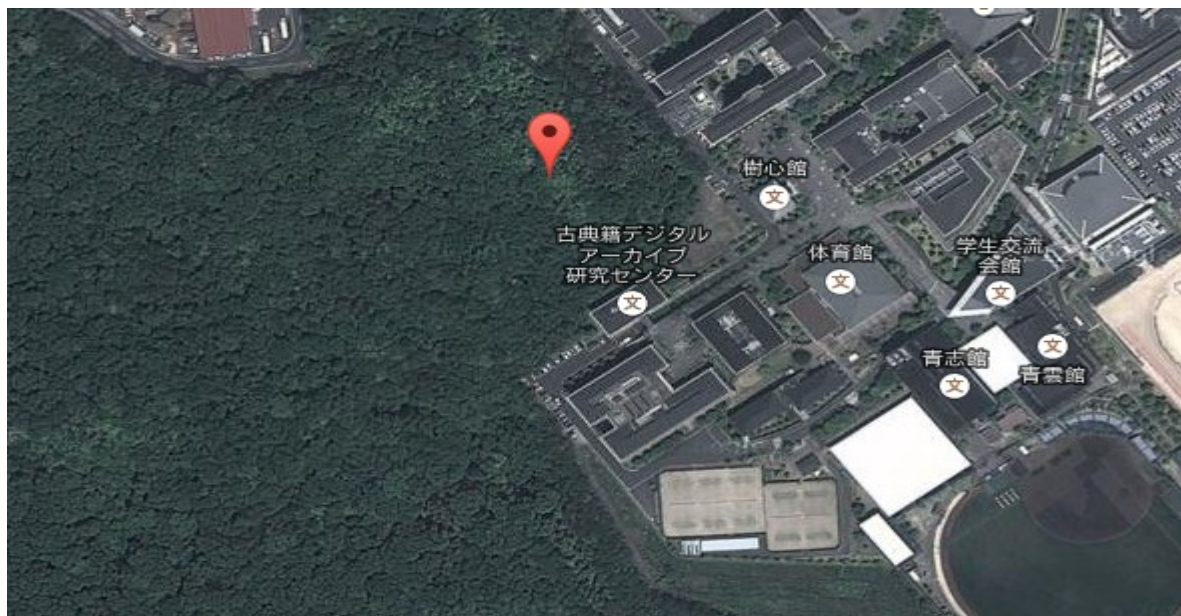


図 12 推定発信源 1

図 12 を見ると、図 11 に示した真の発信源から約 215m 離れていた。

5.2.5 実験目的

測定点を増やしたほうがより正確に中心を求めることができるのではないかと考え、利用する測定点を増やしたときの誤差を調べるために測定点を 11 箇所から 14 箇所に増やして実験を行った。

5.2.6 実験手順

ランダムに 30 箇所測定し、電波強度の値が -75dBm ～ -70dBm という電波強度の値の差が 5dBm の測定点だけを選択し、提案手法により中心を求めた。ただ選択した測定点を 14 箇所にした。

5.2.7「屋外での電波発信源の特定」の実験結果

今回は電波強度が -75dBm ～ -70dBm の測定点だけを選択し中心を求めた。測定点はこれまでの 11 箇所ではなく、14 箇所で中心を求めた。測定結果を表 10 に示す。

表 10 測定結果 2

緯度	経度	dBm
34.9724009034	135.93973196	-72
34.9687335087	135.9401272	-73
34.9626752831	135.93987091	-70
34.9626931075	135.93971936	-71
34.9626674607	135.93974804	-71
34.9627050774	135.93978201	-72
34.9626408496	135.93977755	-74
34.96249932	135.93999092	-72
34.9624362948	135.93991719	-74
34.9627280443	135.93998255	-73
34.962496215	135.94012116	-70
34.9627705075	135.93946633	-75
34.9623502825	135.93964856	-73
34.9625792805	135.9399166	-74

表 2 のデータから提案手法を用いて計算で求めた推定中心を表 11 に示す。

表 11 推定中心 2

x 軸	34.962399542
Y 軸	135.9395752

推定中心をグーグルマップを用いて地図にマッピングした結果を図 13 に示す。



図 13:推定発信源 2

図 13 を見ると、図 11 に示した真の発信源から約 40m 離れている。

5.2.8 考察

以上の実験より考察する。まず図 11 と図 12 を見比べると明らかに違う場所を示していることがわかる。図 11 と図 13 を見比べても距離は減りはしたがまだ誤差がある。このずれは GPS から取得する自分の位置情報に誤差が含まれている場合があるということが考えられる。実際に誤差が含まれていたデータの一例を以下に示す。

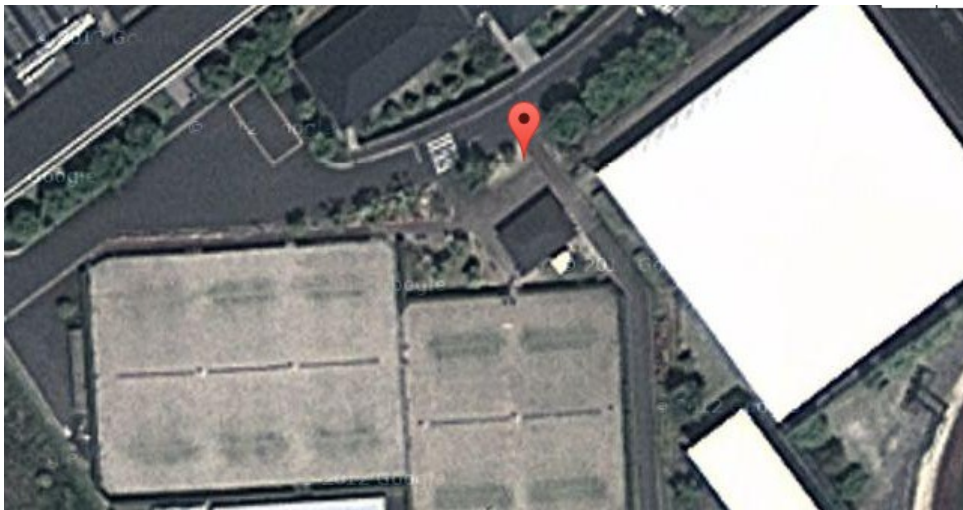


図 14:実際に測定した場所

実際に位置情報を取得した場所を図 14 に示す。しかし、



図 15 GPS から取得した場所

GPS から取得した位置情報をグーグルマップでマッピングした図を図 15 に示す。図 14 図 15 からその差は約 25m ほどである事がわかる。このような誤差があると電波強度の値が実際に計測した値と変わってしまうため、電波強度の差が 10dBm 以内で計測しているつもりでも、実際は 20dBm 以内や 30dBm 以内などになっている可能性が高い。そのために求めたい中心と求めた中心との差につながっているのではないかと考えた。

5.3 シミュレーション実験

なぜ求めたい中心に近づくことができないのかを検討するため、もう一度シミュレーション実験を行う。屋外での実験結果でなぜ計算して出した中心が求めたい結果と同じ場所にならないのか。これは GPS から取得する位置情報に誤差が含まれているのが原因ではないかと考えた。シミュレーション実験を再度行うにあたり、屋外での実験とのマッチングをするため座標で距離の換算を行う。グラフ上での距離は屋外での何 m なのか確認するため、屋外での実験とシミュレーション実験とで電波強度を合わせることで距離の換算を行う。推定中心が(21.2,21.2)の場合、中心からの距離は 1.69 で推定中心の電波強度は-68.3dBm である。屋外で実験した場合-70dBm は中心から約 10m 離れていた。このことからグラフ上の 1.7 は屋外での約 10m にあたる事がわかる。

5.3.1 実験目的

測定点で位置情報を取得するときに誤差を含め、実験を行うと結果にどのような影響がでるのかを確認する。

5.3.2 実験手順

範囲を x 座標を 0 から 40、y 座標 0 から 40 にし、(20,20)に置いた電波の発信源の電波強度を-20dBm とする。電波強度を-20dBm にしたのは屋外の実験で使用したアクセスポイントの電波強度に合わせるためである。範囲の中でランダムに配置した 30 箇所で電波強度を測定した。

その値から電波強度の差が約 10dBm 測定点だけを 11 箇所選択し、最小二乗法で中心を求めるといった実験を行った。

ここで今回は x 座標と y 座標にあらかじめ誤差を含ませることを考えた。予備実験として私が屋外で実験を行った際に GPS から取得する位置情報に誤差が含まれていたのだが、その確立が 50%ほどだった。11 箇所位置情報を取得すると平均して 5,6 箇所の位置情報に誤差が含まれている。そのため、シミュレーションで行う実験でも 5 箇所わざと誤差を含ませる。誤差の値はグラフ上で 2.4、屋外だと約 20m の誤差を含ませる。

前回の実験と同じ数値のものを使う。この表 4 が実際に自分が位置情報を測定した場所とする。

表 12:実際に測定した場所

X	Y	dBm
21.56	21.47	-77.89
17.79	18.53	-83.49
16.62	18.7	-88.35
18.65	22.65	-85.63
17.7	23.36	-89.53
18.4	16.35	-89.34
17.64	18.7	-83.8
23.67	17.82	-89.93
20.28	17.6	-81.32
16.07	18.15	-90.07
20.33	17.84	-78.5

そして、この表 5 が誤差を含めた位置情報である。

表 13:誤差を含めた場所

X	Y	dBm
23.96	23.87	-77.89
15.39	16.13	-83.49
14.22	16.3	-88.35
16.25	25.05	-85.63
15.3	25.76	-89.53
7.087	13.95	-89.34
17.64	18.7	-83.8
23.67	17.82	-89.93
20.28	17.6	-81.32
16.07	18.15	-90.07
20.33	17.84	-78.5

この表 13 を計算し中心を求める。

5.3.3 実験結果

表 14 誤差を含めた結果

X 座標	15.41
Y 座標	19.81

中心からの距離は 4.59 で換算すると、中心(20,20)から約 27m の誤差があることになる。誤差を考慮したシミュレーション実験は屋外での測定結果と一致しているといえる。

第6章 まとめ

シミュレーション実験を3回行った。その結果は電波強度の差が約5dBmの場合と約10dBmの場合、ずれることなく中心を求めることができた。しかし約30dBmの場合、中心からずれる結果となった。それぞれの結果より正確に中心を求めたい場合推測したとおり電波強度の値がより近い測定点のみを選択し計算する必要があると思われる。

シミュレーション実験結果より屋外での実験を行う場合も選択する測定点の電波強度の範囲を10dBm以内にすればより正確に中心の値が求まるのではないかと推測する。しかし5.2節の実験結果の場合、シミュレーションでは問題がなかった電波強度の差が5dBmという状況で実験したのだが求めたい中心の値が計算では求まらずに大きくずれてしまった。そこで測定点の箇所が多ければ多いほどより正確に中心が求まるのではないかと推測した。その結果は実験中心からの誤差は約40mだった。

なぜシミュレーションでは成功したのに、屋外での実験では成功しないのか。その理由はGPSから取得する位置情報に含まれる誤差にあるのではないかと推測し、シミュレーションで最初から位置情報に誤差を含ませる手法を試した。計算で求めた中心は元の中心から約27m離れており、屋外での実験とよく一致した。

災害時にこの手法を使用する場合、素早く建物の下敷きになった人々を救出しなければいけない。そのため誤差はほとんどあってはならないし、1mや2mでも搜索する範囲がずれてしまっていると救出に余計な時間がかかってしまう。天頂衛星を利用するなどGPSの測地精度を向上させる必要がある。

今後の課題としては移動しながらの連続データ記録に対応する事でより測地精度の向上をはかる手法の開発である。

謝辞

本論文作成に当たって日々ご指導頂いた三好力教授には深く感謝いたします。また、様々なアドバイスや相談をくださった同三好研究室の皆様や、友人の皆様には深く感謝いたします。

参考文献

- [1]金谷健一”これなら分かる最適化数学。基礎原理から計算手法まで”
- [2]トランジスタ技術編集部編”GPSのしくみと応用技術。測位原理、受信データの詳細から応用製作まで”
- [3]”エクセルで解く最小二乗法”
<http://imagingsolution.blog107.fc2.com/blog-entry-17.html>
- [4]”デジベル”
<http://www2u.biglobe.ne.jp/~FRN64808/db.htm>
- [5]”wifiのmacアドレスはもはや住所と考えるしかない”
<http://takagi-hiromitsu.jp/diary/20111126.html>
- [6]”Androidプログラミング入門。これ1冊で基礎から応用まで学べます！”

付録

```
package com.taku.gpssample2;

import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;
import android.app.Activity;
import android.content.Context;

public class MainActivity extends Activity
implements LocationListener {

    // 緯度・経度表示用の TextView
    private TextView
    _textViewNetworkLat,
    _textViewNetworkLng, _textViewGpsLat,
    _textViewGpsLng;
    private ProgressBar
    _searchProgressBar;

    private Button
    _buttonSearchNetwork, _buttonSearchGps;

    private Context    _context;
    private LocationManager
    _locationManager;

    // どちらのプロバイダを実行中か
    判別できるようにする
    private int _execSearchProvider;
    private static final int
    EXEC_PROVIDER_NETWORK = 1;
    private static final int
    EXEC_PROVIDER_GPS    = 2;

    @Override
    protected void onCreate(Bundle
```

```
savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);

        _textViewNetworkLat =
    (TextView)findViewById(R.id.textViewNet
    workLat);
        _textViewNetworkLng =
    (TextView)findViewById(R.id.textViewNet
    workLng);
        _textViewGpsLat    =
    (TextView)findViewById(R.id.textViewGps
    Lat);
        _textViewGpsLng    =
    (TextView)findViewById(R.id.textViewGps
    Lng);
        _searchProgressBar =
    (ProgressBar)findViewById(R.id.progressB
    ar_search);
        _buttonSearchNetwork =
    (Button)findViewById(R.id.button_search_l
    ocation_network);
        _buttonSearchGps    =
    (Button)findViewById(R.id.button_search_l
    ocation_gps);
        _context            = this;
        _locationManager    =
    (LocationManager)_context.getSystemService
    (Context.LOCATION_SERVICE);

    // プロGRESSバーは初回非
    表示とする

    _searchProgressBar.setVisibility(View.INVI
    SIBLE);
```

```

_buttonSearchNetwork.setOnClickListener(
(new View.OnClickListener() {
    @Override
    public void
onClick(View v) {
        // ネットワー
クサービスによる検索が有効かチェック
        if
(locationManager.isProviderEnabled(Loca
tionManager.NETWORK_PROVIDER)) {
            locationManager.removeUpdates(MainAct
ivity.this);

            _buttonSearchNetwork.setClickable(false);

            _buttonSearchGps.setClickable(false);

            _searchProgressBar.setVisibility(View.VISI
BLE);

            _execSearchProvider =
EXEC_PROVIDER_NETWORK;
            // 検索を実行

            locationManager.requestLocationUpdates
(LocationManager.NETWORK_PROVIDER
, 0, 0, MainActivity.this);
        } else {
            // 無効

            Toast.makeText(_context, "ネット
ワークサービスが無効のため検索する事が
出来ません",
Toast.LENGTH_SHORT).show();
        }
    }
});

_buttonSearchGps.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void
onClick(View v) {
        // GPS による検
索が有効かチェック
        if
(locationManager.isProviderEnabled(Loca
tionManager.GPS_PROVIDER)) {
            locationManager.removeUpdates(MainAct
ivity.this);

            _buttonSearchNetwork.setClickable(false);

            _buttonSearchGps.setClickable(false);

            _searchProgressBar.setVisibility(View.VISI
BLE);

            _execSearchProvider =
EXEC_PROVIDER_GPS;
            // 検
索を実行

            locationManager.requestLocationUpdates
(LocationManager.GPS_PROVIDER, 0, 0,
MainActivity.this);
        } else {
            // 無効

            Toast.makeText(_context, "GPS が
無効のため検索する事が出来ません",
Toast.LENGTH_SHORT).show();
        }
    }
});

@Override
protected void onPause() {
    super.onPause();
    // バックグラウンドに移動し
た場合は検索を終了させる
}
}

```

```

_locationManager.removeUpdates(this);

_buttonSearchNetwork.setClickable(true);

_buttonSearchGps.setClickable(true);

_searchProgressBar.setVisibility(View.INVISIBLE);
}

@Override
public void
onProviderDisabled(String provider) {
    // TODO Auto-generated
    method stub
}

@Override
public void
onProviderEnabled(String provider) {
    // TODO Auto-generated
    method stub
}

@Override
public void onStatusChanged(String
provider, int status, Bundle extras) {
    // TODO Auto-generated
    method stub
}
}

//public class MainActivity extends
Activity {

_locationManager.removeUpdates(this);
// テンプレートに表示
if (_execSearchProvider ==
EXEC_PROVIDER_GPS) {

_textViewGpsLat.setText(String.valueOf(lo
cation.getLatitude()));

_textViewGpsLng.setText(String.valueOf(l
ocation.getLongitude()));
} else {

_textViewNetworkLat.setText(String.value
Of(location.getLatitude()));

_textViewNetworkLng.setText(String.value
Of(location.getLongitude()));
}
// ボタンを有効に戻す

_buttonSearchNetwork.setClickable(true);

```

```

package com.taku.wifi6;

import java.util.List;
import android.app.ListActivity;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.net.wifi.ScanResult;
import android.net.wifi.WifiConfiguration;
import android.net.wifi.WifiManager;
import android.os.Bundle;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.Toast;
import android.net.wifi.WifiInfo;

public class MainActivity extends
ListActivity {
    @Override
    public void onCreate(Bundle
savedInstanceState) {

super.onCreate(savedInstanceState);
        //setContentView(R.layout.
activity_list_item);

        final WifiManager manager =
(WifiManager)
getService(WIFI_SERVICE);

        // register WiFi scan
results receiver
        IntentFilter filter = new
IntentFilter();

filter.addAction(WifiManager.SCAN_RESU
LTS_AVAILABLE_ACTION);

        registerReceiver(new
BroadcastReceiver() {
            @Override
            public void onReceive(Context context,
Intent arg1) {
                Toast.makeText(context, "receive wifi
scan results",
                Toast.LENGTH_SHORT).show();

                WifiManager
manager =
(WifiManager)getService(WIFI_SER
VICE);

                WifiInfo info =
manager.getConnectionInfo();

                int rssi =
info.getRssi();

                int level =
WifiManager.calculateSignalLevel(rssi, 5);
                //apInfo[3] =
String.format("RSSI : %d / Level : %d/4",
rssi, level);

                List<ScanResult> results =
manager.getScanResults();

                final String[]
items = new String[results.size()];
                for (int i = 0; i
< results.size(); ++i) {

                    items[i] = String.format("SSID:
%s¥nMac アドレス:%s¥n 信号 level:
%sBm¥n",results.get(i).SSID,results.get(i).
BSSID,results.get(i).level);//results.get(i).
SSID + results.get(i).level;
                }

                final

```

```
ArrayAdapter<String> adapter = new
ArrayAdapter<String>(context,
android.R.layout.simple_list_item_1, items);
}
}
setListAdapter(adapter);
}
}, filter);

if (manager.getWifiState() ==
WifiManager.WIFI_STATE_ENABLED) {
// start WiFi Scan
manager.startScan();
}
```