

# センサーネットワークの位相情報の検知に関する研究

成田 龍太\*

徳山 豪†

## 1 はじめに

無線センサーネットワーク (WSNs) は多数の無線機能付き小型センサーデバイスを空間に分布させ、それらが相互に通信を行い、空間の物理的情報を計測することを可能にするネットワークである。

センサーデバイスはバッテリーを電源として駆動している小型の端末であり、また、山や森などの電気を容易に供給できないような場所にも配置されることがある。そのため、WSNs の寿命はセンサーデバイスのバッテリーの駆動時間と同等である。ゆえに、デバイスの駆動時間はネットワークの維持には非常に重要な要素である。

WSNs を利用しているユーザの視点から見れば、観測している領域の境界は重要な要素となる。例えば、森林を観測しているセンサーネットワークで、急に火災が発生し、火災現場にあるセンサーが使えなくなった場合を考える。この場合、センサーが観測できる領域に穴 (hole) が開いてしまい、その地域のデータを観測できなくなってしまう。この穴の付近にあるセンサー (境界ノード) を早く特定できれば火災の早期発見にも繋がる。

また、センサー領域の境界ノードを求める技術は、ジオメトリックルーティングの改善に貢献している。ジオメトリックルーティングとはノードの幾何的な情報を利用してルーティングを行う手法である。ジオメトリックルーティングは経路上のホップ数に対して効率良く行うことができるが、ネットワーク上に障害物が存在すると障害物付近のノードでメッセージが止まってしまい、正しく目的地まで送ることができない問題がある。そのため、障害物を回避してメッセージを送る必要がある。

Rao ら [3] はセンサーの位置が分からないネットワークにおいて、境界ノードの判定が与えられてい

る条件下ですべてのノードの位置を推定するアルゴリズムを提案している。さらに、hole などの障害物が多数配置された場合におけるジオメトリックルーティングでは、仮想座標の場合の方が良い精度でルーティングを行うという結果が得られている。従って、たとえ正確な位置が判っていても、この仮想座標の導出には意義がある。Fang ら [1] はジオメトリックルーティングをする際にメッセージが止まる部分を hole として定義し、通信領域内に存在するノードの位置情報を利用して境界ノード判定を行っている。境界ノードで形成される閉路を hole として構成し、この閉路の情報を予め記憶する。ジオメトリックルーティングでメッセージが止まった場合、閉路上を周回することで障害物から脱出し、メッセージの到達を保証している。

境界ノードを判定する問題は様々な手法で過去に研究されており、Zhang ら [4] は GPS などから得られる位置情報を用いて自分と隣接ノードの位置から境界ノード判定アルゴリズムを提案している。また、Stefan ら [2] は 4~8 ホップ先のノードの接続情報を用いて境界ノード判定を行っている。

境界ノードを判定するために GPS から位置情報を取得することや探索範囲を広げることは非常に有効である。しかし、GPS の電力消費や通信コスト増大によるセンサー寿命の低下につながる。そこで、本研究では、ネットワークの消費電力を抑える観点から GPS などの位置情報取得システムを利用せず、隣接ノードの情報から境界ノードを特定するアルゴリズムを提案する。

## 2 準備

センサーネットワークはセンサーノードと呼ばれる端末が平面上に配置され、無線で通信しあって平面上に構成するネットワークである。従って、セン

\*東北大学大学院情報科学研究科  
†第一著者に同じ

サーノードをノードと呼ばれる平面上の点、通信領域及びセンサー領域を円でモデル化する。

平面上に  $n$  個のノードの集合  $V = \{s_1, s_2, \dots, s_n\}$  が与えられたとする。簡略化のため、センサーノードが存在する領域全体を 2 次元の正方形領域  $A_l = [-l/2, l/2]^2$  とする。また、各ノード  $s_i$  は一定の通信半径  $r_c$  を持つものとする。また、半径が  $r_c$  で中心が  $s_i$  の  $n$  個のディスクの集合  $\mathbb{D} = \{D_1, D_2, \dots, D_n\}$  を考える。  $V$  における無線ネットワークをグラフ  $G(\mathbb{D}) = (V, E)$  で表現する。ここで無向辺  $(s_i, s_j)$  は  $v_i \in D_j$  かつ  $s_j \in D_i$  のとき存在する。すなわち、  $s_i$  と  $s_j$  はお互いの通信半径の中にいる時のみ互いに通信可能であるということである。このようなグラフは単位円グラフ (UDG) と呼ばれる。

平面上のある地点とセンサーノードとのユークリッド距離が  $r_s$  以下であるとき、センサーはその地点を測定できるものとする。そのときの  $r_s$  をセンサーノードのセンシング半径と呼び、各センサーノードで一定の値をもつものとする。ただし、  $2r_s \leq r_c$  が成り立つものとする。ただし、  $2r_s \leq r_c$  が成り立つとする。また、半径が  $r_s$  で中心が  $v_i$  の  $n$  個のディスクの集合を  $\mathbb{B} = \{B_1, B_2, \dots, B_n\}$  とする。

センサーネットワークが観測できない領域を空白領域 (hole) と呼ぶ。hole は平面上の領域として境界を持ち、その境界はセンサー円の円弧で構成される。これらの円弧に対応するノードを境界ノードと言う。まず、以下に hole を定義する。

定義 1.  $A_l$  にノード集合  $S = \{s_1, s_2, \dots, s_n\}$  が配置されたとき、領域

$$\mathbb{O} = \{p \in A_l \mid \forall s \in S, \|p - s\| > r_s\}$$

( $\|p - s\|$  は  $p$  と  $s$  の間のユークリッド距離を表す)

を hole とする。即ち、  $\mathbb{O} = A_l \setminus \bigcup_{i=1}^n B_i$  である。

また、  $\partial\mathbb{O}$  を  $\mathbb{O}$  の境界とする。即ち、  $d(\mathbb{O}) = \{p \in A_l \mid \forall s \in S, \|p - s\| \geq r_s\}$  とすると、  $\partial\mathbb{O} = d(\mathbb{O}) \setminus \mathbb{O}$  である。

定義 2.  $A_l$  にノード集合  $S = \{s_1, s_2, \dots, s_n\}$  が配置されたとき、以下の条件を満たすノード  $s \in S$  を境界ノードと呼ぶ

$$\exists p \in \partial\mathbb{O}, \|p - s\| = r_s$$

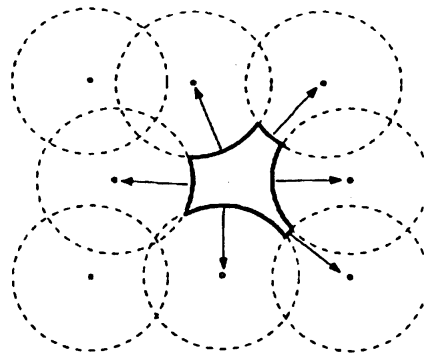


図 1: 境界ノード

図 1 より、hole の境界は、センサー領域の境界でもあり、そこに属しているノードが境界ノードであると言える。また、センサー領域の境界付近は他のノードによって観測されていないということがわかる。

ここで、Zhang ら [4] らはこの境界ノードの定義から、局所ボロノイ領域 (LVP) を利用して境界ノード判定を行った。まず、以下に局所ボロノイ領域の定義を行う。

定義 3. 2 つのノード  $s_i, s_j \in V$  が与えられたとき、  $s_j$  に対する  $s_i$  の支配領域は  $s_j$  より  $s_i$  に近い領域を表し、

$$Dom(s_i, s_j) = \{v \in R \mid \|v - s_i\| \leq \|v - s_j\|\}$$

と定義される。明らかに  $Dom(s_i, s_j)$  は  $s_i$  と  $s_j$  の二等分線により得られる 2 つの半平面のうち、  $s_i$  に近い方の半平面である。

定義 4. ノード  $s_i$  に対する局所ボロノイ領域は、  $s_i$  の各隣接ノード集合  $Neig(s_i)$  に対する  $s_i$  の支配領域の共通領域である。即ち、

$$L(s_i) = \bigcap_{s_j \in Neig(s_i)} Dom(s_i, s_j)$$

と定義する。

境界ノードに対して以下の定理が成り立つ。

定理 1. ある LVP の点  $v \in L(s_i)$  がノード  $s_i$  のセンサー領域内に存在しないならば、  $s_i$  は境界ノードである。ここで、  $r_c \geq 2r_s$  とする。

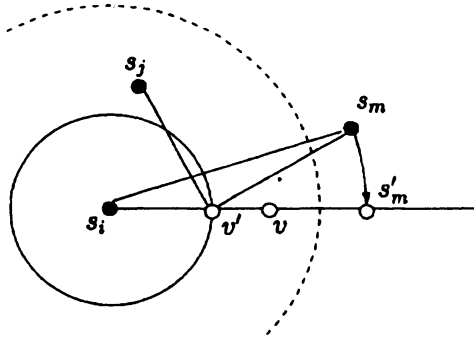


図 2: 定理 1 の証明

証明:  $v \in L(s_i)$  であるので,  $L(s_i)$  の凸性から  $\overline{s_i v}$  と  $s_i$  のセンサー領域の境界との交点  $v'$  も  $v' \in L(s_i)$  である.  $v' \in L(s_i)$  であるので,  $s_i$  の隣接ノード  $s_j$  に対し,  $\|s_j v'\| \geq \|s_i v'\| > r_s$  である. 即ち, どの隣接ノードも  $v'$  を観測することはできない.

次に,  $s_i$  からの非隣接ノード  $s_m$  を考え,  $\|s_i s_m\| = r_c + \delta$ , ( $\delta > 0$ ) とする.  $s'_m$  を  $\|s_i s_m\| = \|s_i s'_m\|$  を満たす  $\overline{s_i v}$  上の点とする. 三角不等式から  $\|s_m v'\| + \|s_i v'\| \geq \|s_i s_m\| = \|s_i s'_m\| = \|s_i v'\| + \|v' s'_m\|$  である. 即ち,  $\|v' s_m\| \geq \|v' s'_m\| = \|s_i s'_m\| - \|s_i v'\| = r_c - r_s + \delta$  となる. ここで,  $r_c \geq 2r_s$  より,  $\|v' s_m\| > r_s$  となる. よって, 非隣接ノードからも  $v'$  をセンシングすることができない. つまり,  $s_i$  はセンシング領域の境界のうち, 他のノードから観測されない地点を持つので境界ノードである.  $\square$

つまり, LVP を作成しセンサー領域内に存在するか判定することで自分が境界ノードかどうか確認できる.

### 3 境界判定アルゴリズム

#### 3.1 入力

センサーノード間の距離を測定する方法として GPS を使用することが考えられるが, 本研究では GPS を使用せずに距離を測定する方法を考える. 通常, 隣接ノードの存在を知るためには通信半径いっぱいメッセージをブロードキャストし, 応答の可否で隣接ノードの存在を知ることができる. この方法では隣接ノードの距離が  $0 \sim r_c$  の範囲内にあると

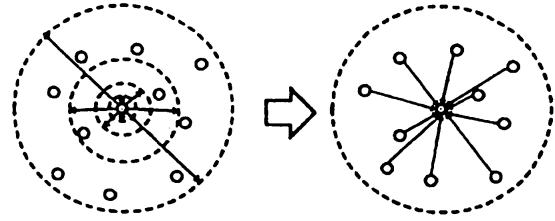


図 3: 通信距離に制限を加えた近傍探索操作

しかわからない. しかし, メッセージを送る通信半径を制限し, 複数回に分けてブロードキャストすることでノード間の距離がある区間内に存在する事を知ることができる (図 3 参照). 本研究ではこのような環境についての境界ノード判定を扱うことにする.

#### 3.2 提案する境界ノード判定手法

##### 3.2.1 座標推定に基づいた境界ノード判定アルゴリズム

我々のアルゴリズムは三辺測量を用いて全てのノードの座標の近似値を再帰的に同定し, LVP を作成して境界ノード判定を行う.

三辺測量は 3 点からなる三角形の各辺の距離を計測し, 各点の位置を測定する技術である. 位置が既知である 2 つの基準点から目的の点までの距離を半径とした円を描き, その交点が測定すべき点の座標となる. しかし, 基準点が 2 つの場合は測定する座標の候補が 2 つ現れるため, 基準点を 3 点に増やし三角測量を行う. 基準点が 3 点ある場合は, 測定する座標の候補が 1 つしか現れないので, 目的の点の座標が一意に決まる.

以上を踏まえて, 以下にノード  $s$  に対する境界ノード判定アルゴリズムの動作を示す. ここで,  $A(s)$  を座標が決定されていないノード集合とする.

座標推定に基づいた境界ノード判定アルゴリズム:

1.  $dist(p, q) \leq r_s$ ,  $p, q \neq s$  を満たす点  $p, q$  を選択
2.  $p, q$  の座標を  $(0, 0)$ ,  $(dist(p, q), 0)$  とする.
3.  $s$  の  $y$  座標が正であると仮定して三辺測量を行う.
4.  $A(s) \leftarrow Neig(s) \setminus \{p, q\}$

5. while  $A(s) \neq \phi$  do
6.   foreach  $A(s)$ に含まれるノード  $t$  do
7.     if  $t$ に決定した隣接ノードが3点以上存在する then
8.       三辺測量を行い  $t$ の座標を決定する
9.        $A(s) \leftarrow A(s) \setminus \{t\}$
10.     end if
11.   end foreach
12. end while
13. LVP を計算し、境界ノードの判定を行う

### 3.2.2 ガブリエルグラフを用いた境界ノード判定アルゴリズム

前述のアルゴリズムでは隣接ノードの位置をすべて求め、LVP を計算するために隣接ノードとの支配領域を計算している。しかし、すべての隣接ノードに対して支配領域を計算する必要はなく、LVP のボロノイ辺を構成しているノードに対して支配領域を計算すれば計算時間の短縮につながる。

どの隣接ノードが LVP のボロノイ辺を構成しているかを調べるには、ドロネー図を考察することで得られる。ドロネー図とはボロノイ図の双対グラフであり、ボロノイセル同士が隣接しているノード対に辺を張ることで構成される。よって、UDG に対してドロネー三角形分割を考えれば LVP の隣接ノードがどれであるか知ることができる。

ドロネー図を構成するには逐次添加法が有名であるが、逐次添加法を行うにはノードの座標を知る必要がある。その為、前節で述べた三辺測量アルゴリズムを用いてすべての座標を求めなければならない。すべての座標を求めていれば LVP が計算できるのでドロネー図を行う必要はない。従って座標を求める点の数の軽減を行う必要がある。

そのため本研究ではドロネー図とは異なり局所的に構築することができるガブリエルグラフを用いる。ガブリエルグラフがガブリエル辺を持つためにはドロネー辺がそれに双対なボロノイ辺と交差することが必要十分条件として知られている。よって、ガブリエル

グラフはドロネー図の部分グラフであり、LVP の隣接ノードでガブリエルグラフで隣接しているものを知ることができ、計算するノードの座標を減らすことができる。しかし、LVP の隣接ノードのうちガブリエルグラフで隣接していないものを求める必要がある。本研究ではその問題点を解決するアルゴリズムを提案する。

ガブリエルグラフを用いた境界ノード判定アルゴリズム：

1. 中心ノードとガブリエルグラフでの隣接点の座標を三辺測量を用いて計算
2. 計算されたノードの座標から LVP を作成
3. if LVP がセンサー領域内にある then
4.   中心ノード  $s$  は非境界ノードとし、終了する
5. else do
6.   センサー領域の境界上の点  $p$  を起点に円周を時計回りに走査し、LVP の領域内に入る時に交差するボロノイ辺に対応する点を  $a$ 、LVP の領域内から出る時に交差するボロノイ辺に対応する点を  $b$  とおく。
7.    $a_0 = s, b_0 = s$  とおく
8.   while true do
9.     foreach  $a, b$  の隣接ノード do
10.       仮想座標座標が計算されていないノードの計算を行う。
11.        $\angle a_0 a a', \angle b_0 b b'$  が最小になる  $a, b$  の隣接ノード  $a', b'$  を選択
12.     end foreach
13.      $a', b'$  を追加し、LVP を再計算
14.     LVP がセンサー領域内に収まれば非境界ノードとし終了
15.      $a_0 = a, b_0 = b$
16.      $a = a', b = b'$

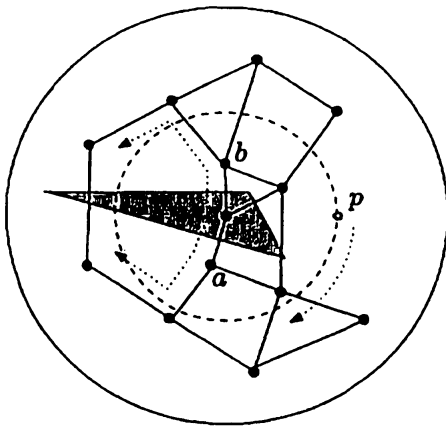


図 4: 探索アルゴリズム

17.  $a', b'$  ともに計算されていない隣接ノードがなければ境界ノードとし終了
18. end while
19. end if

### 3.2.3 画像処理技術を用いた境界ノード判定アルゴリズム

ここでは境界ノード判定法は画像処理のエッジ抽出の理論を利用する。エッジ抽出とは、与えられた画像に対してその画像の輪郭を抽出する技術である。エッジ抽出では自分のピクセルの輝度値と隣接したピクセルの輝度値に対して微分フィルタを用いてエッジ強度を求め、閾値によって自分のピクセルがエッジかどうかを判定している。つまり、自分のピクセルの輝度値と隣接したピクセルの輝度値の差が大きければエッジの要素となる。

これに対し、本研究では通信領域を画像の輝度値と関連付けて画像のエッジ抽出アルゴリズムを応用した。考えとしては hole に近いノードであるほど通信できる領域が狭くなる事を利用する。しかし、ノード自身は近くに hole があるかどうか分かるわけではなく、hole を除く通信領域がどの程度存在するのかわかることができない。そこで、本研究では通信出来る領域を近似的に表す指標としてノードと通信できるノード数を利用した。つまり、このアルゴリズム

ではエッジ抽出アルゴリズムでの輝度値を隣接ノード数として扱う。

隣接ノード数を輝度値として考えると、ノードの密度がネットワーク全体で一様に分布しているならば、隣接ノード数が正しく通信領域の面積と対応しており、エッジ抽出を正しく行うことができる。しかし、ノードの密度のばらつきがある場合は隣接ノード数と通信領域の面積の対応が場所ごとに変化してしまうため、正しく行われぬ可能性がある。そこで、密度を一様とするためにある長さ  $r$  での UDG を考え、そのグラフでの極大独立集合を求める。極大独立集合はランダムにノードを選びマークし、隣接したノードを監視対象から外すという手法を用いれば分散的に容易に求めることができる。

マークされたノードの密度を考えると、マークされたノードの半径  $r$  の領域に他のマークされたノードは存在しない事になるので、マークされたノードに対して分布が一様になる。隣接ノード数を輝度値と設定するのではなく、隣接ノードのうちマークされたノードの数を輝度値として設定すれば正しく通信領域と対応付ける事ができる。

アルゴリズムの流れを次に示す。ここで、 $Count(s_i, I)$  を隣接ノードのうちノード集合  $I$  の要素の個数とする。

画像処理技術を用いた境界ノード判定アルゴリズム:

1. 近傍探索操作により、距離  $r$  と  $r_c$  に分け隣接ノードを計測する
2. 距離  $r$  の UDG から極大独立集合  $I$  を計算する
3. 距離  $r_c$  の UDG から  $Count(s_i, I)$  を計測
4.  $Counts(Neig(s_i), I)$  の情報を隣接ノードから受け取る
5.  $Counts(Neig(s_i), I)$  の平均値  $avg(Counts(Neig(s_i), I))$  を計算する
6.  $L(s_i) = Count(s_i, I) / avg(Counts(Neig(s_i), I))$  をエッジ強度とする
7.  $L(s_i)$  が与えられた閾値  $T$  より小さければ境界ノードと判定する

## 4 アルゴリズムの性能評価

前述のアルゴリズムに対して性能実験を行った。入力環境は  $500 \times 500$  の正方領域にノードを一様に分布させ、領域の中央部分に hole を配置させる。ノードの持つ環境は  $r_c = 35.0, r_s = 17.5$  とする。

ノード間の距離に誤差を与える際、以下の式でノード間の測定距離 *measured\_dist* を定義する。

$$\text{measured\_dist}(s(i), s(j)) = \text{dist}(s(i), s(j)) \times (1 \pm \epsilon)$$

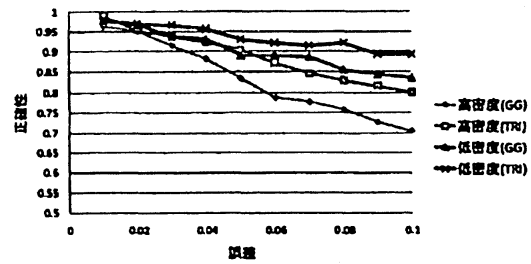
$$(-\text{error} \leq \epsilon \leq \text{error})$$

計測されるノード間距離は実際の距離に比例した誤差を付加して与えられるものとする。実験では *error* の値を変化させることで、アルゴリズムの性能比較を行っている。

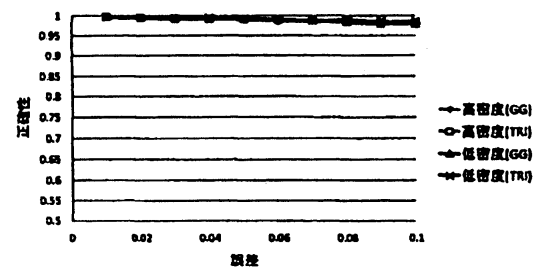
まず、座標推定に基づいた境界ノード判定アルゴリズム (TRI) とガブリエルグラフを用いた境界ノード判定アルゴリズム (GG) の性能実験を行った。図 5 に各アルゴリズムの精度を示す。誤差が増えるたびに境界ノードの認識率は下がっているが非境界ノードの認識率はあまり変化しない傾向が見られる。誤差が増えると仮想座標上のノードの位置に広がりが見える傾向が強くなる。そのため、非境界ノードは隣接ノードの配置が一様であるので認識率はあまり変化しない。しかし、境界ノードだった場合は隣接ノードの配置が偏っているため、誤差の影響で仮想座標が広がってしまい、LVP がセンサー領域に収まってしまうことがある。そのため認識率が下がってしまう傾向になると考えられる。また、GG の場合は TRI と比べ同定するノード数が少ないので全体的な認識率が下がってしまったものだと考えられる。

図 5(a) より、ノードの密度を下げると双方のアルゴリズムとも精度が大きく上がっていることがわかる。通信領域内に存在するノード数が少ないほどノード間の平均距離が大きくなるため、三辺測量による誤差が抑えられたのが要因と考えられる。また、図 5(b) より境界でないノードの判定では密度が変わっても大きな変化が見られなかった。ノードが一様に分散しているため、三辺測量における誤差の影響が密度によって大きく変化しないためだと考えられる。

次に、図 6 に画像処理技術を用いた境界ノード判定アルゴリズムにおいて閾値を選んだ場合の境界ノードと非境界ノードの認識率の分布を示す。r が長い



(a) 境界ノードの認識率



(b) 非境界ノードの認識率

図 5: 座標推定に基づいた境界ノードアルゴリズムの精度

場合は境界でないノードの認識率が下がり始める閾値が低く、境界ノードの認識率はほぼ線形に上がっているため、両方の認識率が良くなるような閾値が存在しない結果となる。r が短い場合は境界でないノードの認識率が下がり始める閾値が 0.8 程度と高く、境界ノードの認識率は閾値が 0.8 辺りを境に伸びが緩やかになっている。つまり、閾値として 0.8 付近を選択すれば、境界ノード及び境界でないノードの認識率が共に高くなる結果が得られた。

境界でないノードが境界ノードと誤認識した場合、誤認識した部分に対して hole が形成されてしまう。これはセンサーが一様に分布する領域を hole と誤認識しているため、極力避けるべきである。一方、境界ノードを境界でないノードと誤認識した場合は、誤認識した割合があまり多くなければ、境界ノードの集合を多角形に構成することで近似的に hole を形成できる。そのため、低い割合の誤認識は許容できる。座標推定による境界判定アルゴリズムでは境界で

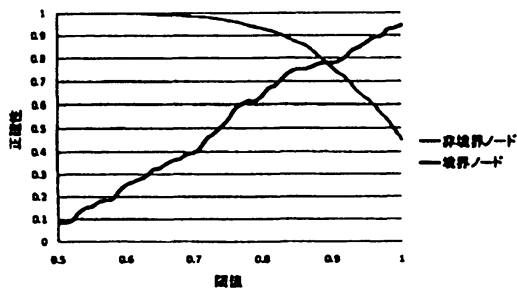
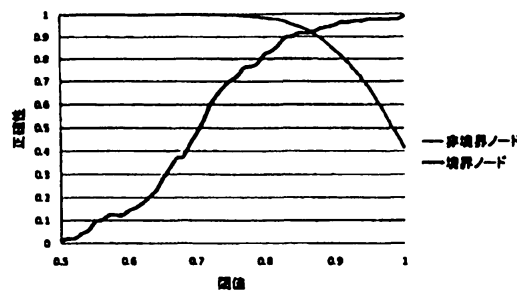
(a)  $r = 14$ (b)  $r = 7$ 

図 6: 認識率の分布

ないノードに対しては非常に高い認識率で、距離誤差が 10% ある場合は境界ノードは 7~8 割認識する結果が得られた。これは境界を多角形として構成するには十分な認識率である。

また、画像処理手法に関してはアルゴリズムは高速であるが、境界でないノードに対する認識率を座標推定によるアルゴリズムと同じになるように閾値を設定すると、 $r = 14$  の場合では境界ノードの認識率の精度が劣ってしまう。しかし、 $r = 7$  の場合は境界ノードの認識率が 8 割と座標推定によるアルゴリズムと同様の結果が得られた。一様にセンサーノードが分布している環境下では  $r$  を短くすることにより、良い精度の結果が得られる。

## 5 まとめ

本研究ではセンサーネットワーク上で境界ノード判定アルゴリズムを座標推定とノード密度による判定手法の 2 種類を提案した。

性能実験においては、座標推定による境界ノード判定アルゴリズムでは、距離誤差を 10% 許した場合においても、非境界頂点の誤認識はほとんどなく、境界ノードの 7~8 割は認識する。これは境界を多角形として構成するには十分な認識率である。また、低密度な配置であるほど高い精度が得られることが確認できた。一方、画像処理手法の方がアルゴリズムは高速であるが、境界でないノードの認識率を座標推定による境界判定アルゴリズムと同程度にすると、 $r$  が大きい場合は座標推定手法と比べ精度は劣る。しかし、 $r$  の値を小さくし、極大独立集合の密度を上げることで高い精度を得ることができた。

## 参考文献

- [1] Q. Fang, J. Gao, and L. J. Guibas. Locating and bypassing routing holes in sensor networks. In *23rd Conf. of the IEEE Communications Society (INFOCOM)*, 2004.
- [2] S. Funke and C. Klein. Hole detection or: "how much geometry hides in connectivity?". In *Symposium on Computational Geometry*, pages 377-385, 2006.
- [3] A. Rao, C. H. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location information. In *Proceedings of Ninth Annual International Conference on Mobile Computing and Networking (MOBICOM)*, pages 96-108, 2003.
- [4] C. Zhang, Y. Zhang, and Y. Fang. Localized algorithms for coverage boundary detection in wireless sensor networks. *Wireless Networks*, 15(1):3-20, 2009.