

# 密なモバイルセンサネットワークにおける センサデータの傍受を用いた効率的な境界線検出手法

松尾 和哉<sup>1</sup> 後藤 啓介<sup>1</sup> 神崎 映光<sup>1</sup> 原 隆浩<sup>1</sup> 西尾 章治郎<sup>1</sup>

**概要：**移動型センサ端末が密に存在するモバイルセンサネットワークにおいて、アプリケーションが観測値の地理的な境界線を要求する場合、境界線付近の端末のみからセンサデータを収集することで、トラフィックを削減しつつ、アプリケーションの要求を満足できる。これまでに、効率的な境界線検出手法は多数提案されているが、それらは端末の位置が固定されており、各端末が隣接端末の位置を把握している環境を想定している。そのため、それらの既存手法を端末が移動する環境で用いた場合、隣接端末の位置の把握に要するトラフィックが大量に発生する。そこで本稿では、センサデータの傍受を用いた効率的な境界線検出手法を提案する。提案手法では、各端末が通信の傍受によって得た隣接端末のセンサデータを基に、自身が境界線付近に位置するかどうかを判断する。このとき、境界線に近い端末ほど優先的にセンサデータを送信することで、境界線検出に要するトラフィックを削減する。さらに本稿では、提案手法の性能評価をシミュレーション実験によって検証する。

## 1. はじめに

近年、一般ユーザの携帯するスマートフォンなどの移動型センサ端末により、センサデータを収集するユーザ参加型センシングが注目されている [2]。ユーザ参加型センシングにおけるデータ収集方法としては、携帯電話回線などのインフラを用いてインターネットに接続しているデータセンタにセンサデータをアップロードすることが考えられる。しかし、インフラの通信帯域は多数のアプリケーションが共用しており、ユーザ参加型センシングが大量のトラフィックを生成して、有限な通信帯域を圧迫することは好ましくない。また、輻輳などによりインフラに一時的な障害が発生している状況でもセンサデータを収集できることが望ましい。そのため、端末間の直接通信によりネットワークを構築し、センサデータを収集するモバイルセンサネットワークが注目されている [10]。モバイルセンサネットワークでは、端末は温度や音といった物理現象をセンシングし、端末間の無線マルチホップ通信を用いて、センサデータを収集するシンクへ送信する。

ここで、一般ユーザの携帯端末で構成されるモバイルセンサネットワークでは、端末数が非常に多くなり、直接通信可能な端末（隣接端末）が常に多数存在する（これを密なモバイルセンサネットワークと呼ぶ）。一方、気温や騒音

などの環境情報は、観測値が一定の範囲に収まる領域など、観測値に基づいた空間的な領域の分類が重要となる場合が多い。例えば、アプリケーションが観測領域内における観測値に基づいた地理的な境界線を要求する場合、シンクは端末から収集したセンサデータを用いて境界線を推定する。このとき、境界線から地理的に遠い端末のセンサデータは必要なく、境界線付近の端末が所持するセンサデータを収集できれば、高い精度で境界線を推定できる。

上記を実現する研究は多数存在するが [9]、それらの研究で提案されている手法は、端末の位置が固定されている環境を想定しており、境界線付近の端末を識別する（これを境界線の検出と呼ぶ）ためには、各端末が隣接端末の位置情報を常に把握している必要がある。しかし、本研究で想定するモバイルセンサネットワークでは、各端末が自由に移動しており、各端末が自身の隣接端末の位置情報を常に把握するためには、すべての端末が隣接端末と位置情報を交換する必要がある。そのため、隣接端末数が多い環境では、トラフィックが著しく増大する。

本稿では、センサデータの傍受により効率的に境界線を検出する手法として、OBD (Overhearing based Boundary Detection) を提案する。OBD では、境界線検出時にのみ、一部の端末間で、観測値と位置情報を含むセンサデータの交換を行う。具体的には、各端末が自身および傍受したセンサデータを用いてダイアグラム（ボロノイ図またはガブリエルグラフ）を作成する。観測値の分類が異なる端末と

<sup>1</sup> 大阪大学 大学院情報科学研究科 マルチメディア工学専攻  
{matsuo.kazuya, goto.keisuke, kanzaki, hara, nishio}@ist.osaka-u.ac.jp

ダイアグラム上で隣接する各端末が、自身のセンサデータを送信する。ここで OBD では、境界線に近い端末ほど優先的にセンサデータを送信するようにする。その他の端末は、送信されたセンサデータを傍受し、自身の周辺におけるダイアグラムを作成することで、自身が境界線から遠いことを認識し、センサデータの送信を中止する。これにより、境界線付近の端末のみがセンサデータを送信できるため、境界線検出に要するトラフィックを削減できる。

以下では、まず 2 章で想定環境について述べ、3 章で本研究で用いるダイアグラムについて述べる。4 章で関連研究について述べ、5 章で提案手法について述べた後、6 章でシミュレーション実験の結果を示す。最後に 7 章で本稿のまとめと今後の課題について述べる。

## 2. 想定環境

本研究では、一般ユーザの携帯する無線通信機能を搭載した移動型センサ端末が、互いに無線マルチホップ通信で接続している密なモバイルセンサネットワークを想定する。各端末は温度などの物理現象を観測可能なデバイスを搭載しているが、アプリケーションが要求する観測値や観測領域といったデータ収集に関する情報はもっておらず、後述するクエリを受信した時点でセンシングを行う。各端末の無線通信範囲は等しく、半径  $r$  の円とする。各端末は GPS などの測地装置を備えており、任意の時刻に自身の位置情報を取得できる。各端末は自由に移動するが、シンクの位置は固定されているものとする。

対象とする観測領域は 2 次元平面を想定する。シンクは観測値の閾値を設定し、閾値未満の観測値を示す領域と閾値以上の観測値を示す領域の境を推定するために、センサデータ（観測値とセンシング時の位置情報）を収集する。以降では、閾値により地理的に領域を分類する線を、観測値の境界線と定義する。また、境界線によって分類された領域において、互いに異なる領域に存在する端末がセンシングした観測値を、異なる観測値と表現する。

## 3. 前提知識

後述する関連研究を含め、境界線を検出する手法では、端末を母点としたボロノイ図を用いることが多く、提案手法でもこれを用いた境界線検出を行う。また提案手法では、境界線検出時のさらなるトラフィック削減を目的に、ボロノイ図の双対であるドロネーグラフの部分グラフであるガブリエルグラフも用いる。本章では、これらのダイアグラムについて説明する。

### 3.1 ボロノイ図

ボロノイ図とは、平面上の任意の点を最も近い母点に属する領域（ボロノイ領域）として分割した図であり（図 1）、互いのボロノイ領域が隣接する（ボロノイ領域上で隣接す

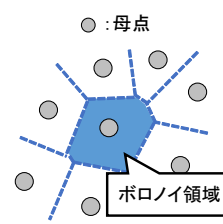


図 1 ボロノイ図

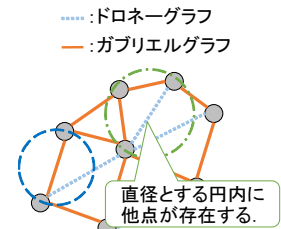


図 2 ガブリエルグラフ

る）二つの母点は、他の母点より近傍に存在する。そのため、各端末を母点とするボロノイ図において、ボロノイ領域上で隣接する端末の観測値が異なる場合、それらの端末は境界線に十分近い位置に存在すると判断できる。

### 3.2 ガブリエルグラフ

ガブリエルグラフとは、ボロノイ領域の隣接関係を示すドロネーグラフの部分グラフであり、平面上に分布する点群に対して、任意の 2 点を結ぶ線分を直径とする円内に他の点が存在しないように点同士を結んだグラフである（図 2）。各端末を頂点とするガブリエルグラフでは、ボロノイ領域上で隣接する端末のうち、互いの距離が短いものが優先的に隣接するため、各端末が隣接する端末数がボロノイ領域上で隣接する端末数以下になる。ガブリエルグラフ上で隣接する端末の観測値が異なる場合も、境界線に十分近い位置に存在すると判断できるが、これらの端末は、ボロノイ領域上で観測値が異なる端末と隣接する端末のうち、境界線により近い位置に存在するもののみに限定される。

## 4. 関連研究

文献 [3], [4] では、観測値を離散化した環境において、固定センサネットワークを形成する端末を母点としたボロノイ図を用いて観測値の境界線を検出する手法を提案している。文献 [3] で提案されている手法では、観測値が互いに異なる母点によって形成されるボロノイ領域の境界（ボロノイ境界）を観測値の境界線として検出している。また文献 [4] では、ボロノイ図を用いて検出した境界線の時間的な変化にも対応できる手法を提案している。これらの研究は、ボロノイ図を用いて観測値の境界線を検出している点で本研究と類似しているが、端末の位置が固定されており、ネットワーク内の端末を母点としたボロノイ図があらかじめ完成している点で本研究と異なる。

端末が密に配置されている環境において、観測値の等値線を検出するアプリケーションを想定した研究も存在する [5], [8]。文献 [5] で提案されている手法では、等値線の値に近い観測値を所持する端末が、自身の隣接端末から収集した観測値から計算される観測値の傾きをシンクへ返信する。文献 [8] で提案されている手法では、等値線付近で転送木を構築し、等値線をまたぐ枝をもつ端末のみからセン

サデータを収集することで、等値線検出のためのトラヒックを抑えている。これらの研究は、端末が密に配置されている環境で等値線を検出する際に発生するトラヒックを削減する目的が本研究と類似するが、文献 [3], [4] と同様に、固定センサネットワークを想定しており、すべての端末が隣接端末の情報を把握している必要がある。そのため、本研究の想定環境にそのまま適用するとトラヒックが大量に発生する。

さらに、いずれの手法も、境界線や等値線を検出するために専用の端末を配置しており、一般ユーザの携帯する端末を利用する本研究とは想定が異なる。

## 5. 提案手法：OBD

OBD では、5.1 節で述べる方法で観測領域全体にクエリを伝播し、5.2 節で述べる方法で境界線を検出する。境界線検出時に送信されたセンサデータは、5.3 節で述べる方法で収集する。

以下では、それぞれの手順について詳述する。

### 5.1 クエリ伝播

2 章で述べたとおり、本稿で想定する端末は、データ収集に関する情報をもっていない。そこで、まずシンクは、これらの情報をクエリとして観測領域内に伝播する。

端末およびシンクは、文献 [6] で提案された手法を基にした位置依存フラッディングプロトコルに従い、必要最小限のトラヒックで観測領域全体にクエリを転送する。クエリには、データ収集に関する情報として、観測領域、観測値の種類、観測値の閾値を含む。また、クエリを転送した端末の情報として、転送端末の識別子、位置、取得した観測値を含む。さらに、後述する境界線検出処理の終了条件として、センサデータ収集開始時刻を含む。

まずシンクはアプリケーションからの要求に応じたクエリを生成し、自身の通信範囲内に転送する。クエリを受信した各端末はセンシングを行った後、待ち時間を設けてクエリを転送する。この待ち時間は、受信したクエリの転送元端末から遠いほど短く設定される。待ち時間内に他の端末が転送したクエリを傍受し、他のクエリ転送端末の通信範囲によって、クエリが指定する観測領域と自身の通信範囲が重なる領域の全体が被覆された場合、クエリの転送を中止する。待ち時間内に該当領域が被覆されなかった端末は、クエリ内の転送端末に関する情報を更新し、クエリを転送する。また、クエリを既に受信している端末および観測領域外に存在する端末は、受信したクエリを破棄し、クエリの転送は行わない。これにより、必要最小限の転送回数で観測領域全体にクエリを伝播できる。

また、クエリを受信した各端末は、クエリ転送端末の識別子を記録し、この情報を用いてシンクを根とする転送木を構築する。センサデータ収集時には、この転送木を用い

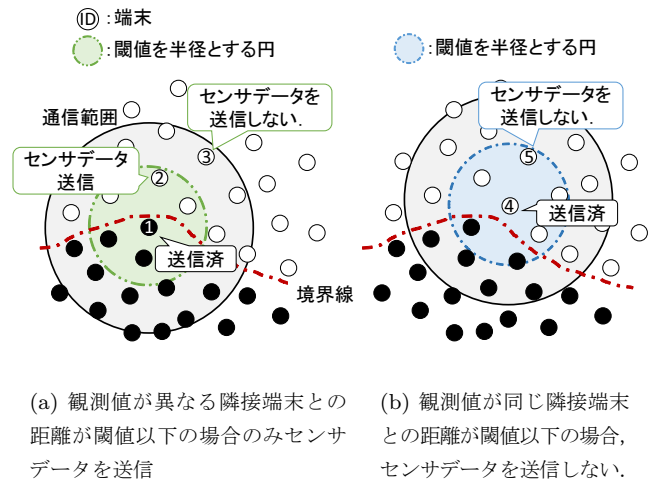


図3 境界線から地理的に遠い端末のセンサデータの送信を抑制するための単純な条件

てシンクまでセンサデータを転送する。

さらに、各端末は、受信したクエリ内に含まれる観測値と自身の観測値を比較する。これらの観測値が異なる場合、自身の通信範囲内に境界線が存在する、すなわち自身が境界線付近に存在する可能性があることを認識し、次節で述べる境界線の検出処理を開始する。このように OBD では、本稿で想定する環境において必要となるクエリ伝播を利用し、境界線にある程度近い範囲内でのみ境界線検出処理を行う。これにより、境界線から遠い領域での情報交換が不要となり、トラヒックの増大が抑制できる。

### 5.2 境界線の検出

5.1 節で述べたクエリの伝播により、クエリ転送端末と観測値が異なる全端末が、自身の通信範囲内に境界線が存在することを認識できる。これらの端末すべてがセンサデータを送信すれば、そのセンサデータに含まれる観測値と異なる観測値をもつ端末も、自身の通信範囲内に境界線が存在することを認識できる。この処理を繰り返すことにより、通信範囲内に境界線が存在する全端末がセンサデータを送信でき、結果として境界線を検出できる。しかし、密なモバイルセンサネットワークでは、境界線から地理的に遠い端末までセンサデータを送信してしまい、無駄なトラヒックが発生する。これを抑制するためには、図3のように、センサデータを送信する条件を、端末間の距離に基づいて設定することが有効である。図において、互いに色が異なる端末は観測値が異なることを示す。まず、単純なものとして、図3(a)のように、観測値が異なる隣接端末との距離が閾値以下の場合のみセンサデータを送信することが考えられる。この閾値を通信範囲より小さく設定することで、図中の端末3のように、境界線から遠い端末のセンサデータの送信を抑制できる。しかし、この閾値を小さくし過ぎると、境界線に十分近いにも関わらず、センサデー

タを送信しない端末が現れ、結果として境界線が検出できなくなる可能性がある。また、境界線から地理的に遠い端末のセンサデータの送信を抑制するための条件としては、図 3(b) における端末 5 のように、観測値が同じ隣接端末との距離が閾値以下の場合、センサデータを送信しないことも考えられる。この閾値を大きく設定することで、トラヒックの削減が期待できるが、単純に観測値が同じ端末との距離のみからセンサデータを送信するかどうかを判断するため、境界線に十分近い位置に存在する端末までセンサデータを送信できなくなる可能性がある。

そこで OBD では、ボロノイ図またはガブリエルグラフを用いて、センサデータの送信を境界線に十分近い位置に存在する端末のみに限定する。具体的には、各端末が、傍受したセンサデータを用いて、自身の通信範囲内におけるボロノイ図またはガブリエルグラフを作成し、前者を用いる OBD-V (OBD using Voronoi diagram) ではボロノイ領域、後者を用いる OBD-G (OBD using Gabriel graph) ではガブリエルグラフ上で、観測値が異なる端末と隣接する場合にのみセンサデータを送信する。また、センサデータの送信は、ダイアグラム上で隣接し、かつ観測値が異なる端末との距離が小さいほど短く設定されるタイマに基づいて行う。これにより、境界線から遠い端末のセンサデータの送信を抑制できる。以降では、ボロノイ領域やガブリエルグラフ上で観測値が異なる端末と隣接する端末を、境界線に接する端末と表現する。

境界線検出処理における端末  $n$  の動作を Algorithm1 に示す。観測値が異なるクエリ  $Q_{dif}$  またはセンサデータを受信した端末  $n$  は、以下の手順に従って動作する。

- (1) クエリまたはセンサデータに含まれる位置情報および自身の位置情報を基にダイアグラム  $D(n)$  を作成する (7, 19 行目)。
- (2) ダイアグラム上で隣接し、かつ観測値の異なる端末  $DN(n)$  が存在し、さらに  $DN(n)$  の中で最も自身に近い端末  $CDN(n)$  が、手順 (1) を実行する前の  $CDN(n)$  ( $CDN_{prev}(n)$ ) と異なる場合 (23 行目)、次式により、自身のセンサデータを送信するまでのタイマ  $T$  を設定する (24 行目)。

$$T = \frac{d}{r} \cdot Max\_delay. \quad (1)$$

上式の  $d$  は  $CDN(n)$  との距離、 $Max\_delay$  はタイマ  $T$  の最大値である。式 (1) より、タイマ  $T$  は、 $CDN(n)$  との距離が小さいほど短く設定される。なお、 $Q_{dif}$  を受信した場合も、式 (1) により、タイマ  $T$  を設定する (10 行目)。

- (3) タイマが切れるまでに  $CDN(n)$  に変更がなかった場合、自身のセンサデータを、5.1 節で構築した転送木上の親端末へ送信し、処理を終了する (14 行目)。
- (4)  $DN(n)$  が存在しない場合、センサデータの送信は行

---

**Algorithm 1** OBD における境界線検出

---

```

1:  $t_{gather} \leftarrow$  gathering starting time specified
   in the received query
   // センサデータ収集開始時刻
2:  $DN(n) \leftarrow \{\emptyset\}$ 
   // ダイアグラム上で隣接し、観測値が異なる端末群
3:  $CDN(n) \leftarrow \emptyset$  //  $DN(n)$  の中で最も端末  $n$  に近い端末
4:  $CDN_{prev}(n) \leftarrow \emptyset$  // センサデータ傍受前の  $CDN(n)$ 
5:  $T \leftarrow \infty$  // タイマ
6: if Received query is  $Q_{dif}$  then
7:   Create  $D(n)$ .
8:    $DN(n) \leftarrow$  the node which sent  $Q_{dif}$ 
9:    $CDN(n) \leftarrow$  the node which sent  $Q_{dif}$ 
10:  Set  $T$ .
11: end if
12: while  $t_{gather}$  does not come do
13:   if Timer  $T$  expired then
14:     Send its sensor data to its parent on the forward-
       ing tree.
15:     break
16:   end if
17:   if Overheard sensor data then
18:      $CDN_{prev}(n) \leftarrow CDN(n)$ 
19:     Create  $D(n)$ .
20:      $DN(n) \leftarrow$  Neighboring nodes with different
       sensor reading in  $D(n)$ 
21:     if  $DN(n) \neq \emptyset$  then
22:        $CDN(n) \leftarrow$  Closest node in  $DN(n)$ 
23:       if  $CDN(n) \neq CDN_{prev}(n)$  then
24:         Set  $T$ .
25:       end if
26:     else
27:        $CDN(n) \leftarrow \emptyset$ 
28:        $T \leftarrow \infty$  // タイマ解除
29:     end if
30:   end if
31: end while

```

---

わない。また、新たに傍受したセンサデータによって  $DN(n)$  が存在しなくなった場合、センサデータの送信を中止する (28 行目)。

- (5) シンクが設定したセンサデータ収集開始時刻  $t_{gather}$  までに新たにセンサデータを傍受した場合は、手順 (1) に戻る (17 行目)。

上記の手順 (1) において作成するダイアグラム  $D(n)$  は、OBD-V ではボロノイ図、OBD-G ではガブリエルグラフとなる。以下では、それぞれの詳細について述べる。

### 5.2.1 OBD-V

OBD-V における各端末の動作として、図 4 のように、端末 7 が送信したクエリを、端末  $\{1, \dots, 6\}$  が受信した場合を想定した例を図 5 に示す。図 5 において、色が薄い端末はセンサデータを送信していない端末を示す。また、これらの端末は互いに直接通信可能であり、互いが送信したセンサデータを傍受可能であるものとする。

端末  $\{1, \dots, 6\}$  は、まず端末 7 を転送木上の親として記録し、自身と端末 7 を母点としたボロノイ図を作成する。例えば図 5(a) は、端末 1 が作成したボロノイ図である。図からわかるとおり、端末 1 はボロノイ領域上で端末 7 と隣



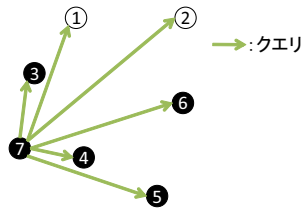


図 4 クエリ転送の例

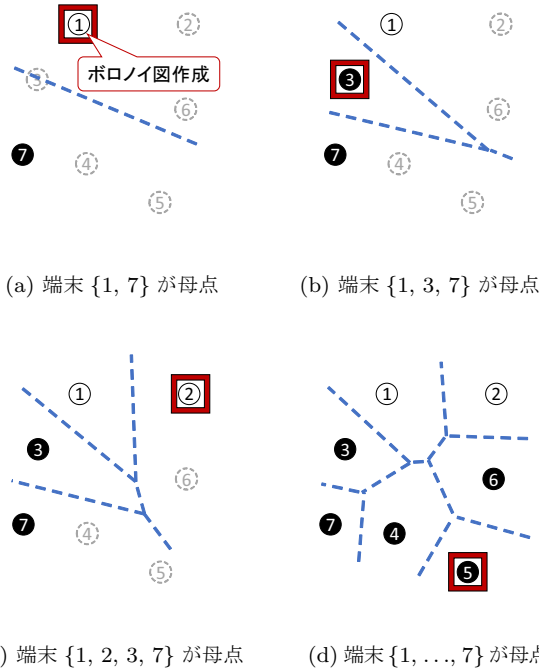
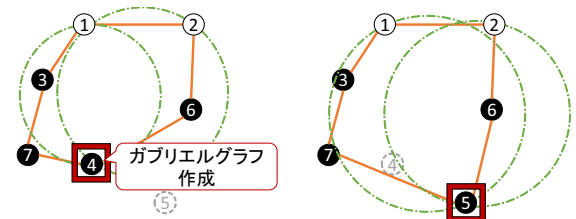


図 5 OBD-V の動作例

接し、互いの観測値が異なるため、端末 7 との距離に応じたタイマを設定する。端末 2 も同様に、端末 7 との距離に応じたタイマを設定する。ここで、端末 2 より端末 1 のほうが端末 7 に近いため、端末 1 のタイマが先に切れ、転送木上の親である端末 7 へセンサデータを送信する。このセンサデータを傍受した端末  $\{2, \dots, 7\}$  は、端末 1 の存在および観測値を認識し、この時点で傍受している端末  $\{1, 7\}$  のセンサデータを基に、端末  $\{1, 7\}$  と自身を母点とするボロノイ図を作成する。例えば端末 3 は、図 5(b) に示すボロノイ図を作成し、自身と観測値が異なる端末 1 と隣接するため、端末 1 との距離に応じたタイマを設定する。端末  $\{4, \dots, 7\}$  も端末 1 との距離に応じたタイマを設定するが、最も端末 1 に近い端末 3 が、端末 7 へセンサデータを送信する。端末 3 のセンサデータを傍受した端末のうち、センサデータを送信していない端末  $\{2, 4, \dots, 7\}$  は、自身と端末  $\{1, 3, 7\}$  を母点とするボロノイ図を作成する。例えば端末 2 は、図 5(c) に示すボロノイ図を作成する。ここで、端末 2 はボロノイ領域上で自身と隣接し、かつ観測値の異なる端末  $DN(2) = \{3, 7\}$  の中で、自身に最も近い端末  $CDN(2)$  が端末 7 から端末 3 に変わるため、端末 3 と



(a) 端末 4 が作成したグラフ

(b) 端末 5 が作成したグラフ

図 6 OBD-G の動作例

の距離に応じたタイマを再設定する。この手順が繰り返されることで、さらに端末  $\{2, 4, 6\}$  が端末 7 へセンサデータを送信する。これらの端末のセンサデータを傍受した端末 5 は、図 5(d) に示すボロノイ図を作成する。図からわかるとおり、端末 5 がボロノイ領域上で隣接する端末は  $\{4, 6\}$  であり、観測値が異なる端末が存在しない ( $DN(5) = \emptyset$ ) ため、端末 5 はセンサデータの送信を中止する。

### 5.2.2 OBD-G

OBD-G における各端末の動作として、前項と同様、図 4 の端末 7 が送信したクエリを、端末  $\{1, \dots, 6\}$  が受信した場合を想定した例を図 6 に示す。図では、クエリ受信後、端末 1、端末 3、端末 6、端末 2 の順にセンサデータの送信が完了している。

センサデータを送信していない端末のうち、端末  $\{4, 5\}$  は、この時点で傍受している端末  $\{1, 2, 3, 6, 7\}$  のセンサデータを基に、それぞれ図 6(a)、図 6(b) に示すガブリエルグラフを作成する。図からわかるとおり、これらの端末は、観測値が異なる端末  $\{1, 2\}$  とガブリエルグラフ上で隣接しない ( $DN(4) = \emptyset$ ,  $DN(5) = \emptyset$ ) ため、センサデータの送信は行わない。ここで、OBD-V においてセンサデータを送信する端末 4 が、OBD-G では送信しない。このように OBD-G では、センサデータを送信する端末数が、OBD-V を用いた場合より少なくなり、境界線検出時のトラヒックをさらに削減できる。

### 5.3 センサデータ収集

センサデータ収集開始時刻になった時点で、5.1 節で述べた転送木の節点に該当する端末は収集用の待ち時間を設定し、待ち時間経過後に、それまでに受信したセンサデータをひとつの packets にまとめて、転送木上の親端末へ送信する。パケットを受信した親端末は、子端末へ ACK を送信する。ACK を受信しなかった子端末は、親端末の ACK が受信できるまでパケットを再送する。親端末がパケットを送信した後に、自身の子端末からパケットを受信した場合は、そのパケットをそのまま自身の親端末へ送信する。ここで設定される収集用の待ち時間は、シンクからのホップ数が多いほど短く設定される。これにより、シンクから

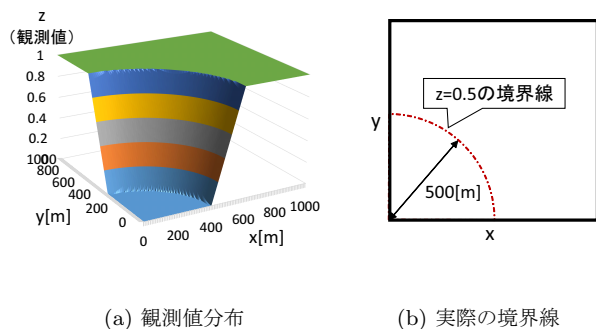


図 7 評価に用いた観測値分布および境界線

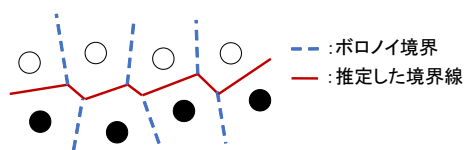


図 8 シンクによる境界線の推定

遠い端末から順にパケットが送信され、最終的に、観測領域内で送信された全センサデータが、シンクに収集される。

## 6. 性能評価

本章では、提案手法の性能評価のために行ったシミュレーション実験の結果を示す。本実験では、ネットワークシミュレータ Scenargie 1.6[7] を用いた。

### 6.1 シミュレーション環境

1,000[m]×1,000[m] の 2 次元平面上に、 $N$  ( $1,000 \leq N \leq 5,000$ ) 台の移動型センサ端末および 1 台のシンクが存在するものとした。シンクは、領域左端および下端から (50[m], 50[m]) の座標に配置した。各端末はランダムウェイポイントモデル [1] に従い、[0.5, 1.0][m/s] の速度で等速移動し、目標位置で 60[s] 停止する。端末間の通信には IEEE 802.11g を使用し、伝送速度 6[Mbps]、 $r = 100$ [m] 程度となる送信電力でパケットを送信する。センサデータ収集開始時刻  $t_{gather}$  は、クエリ発行時刻から 1[s] 後とした。

図 7(a) に、本実験で用いた観測値分布を示す。シンクは観測値の閾値を 0.5 として境界線を要求する。このとき、全体の観測値分布が把握できたときの境界線は、図 7(b) に示す円弧となる。本章では、この円弧を境界線の正解とし、実際の境界線と呼ぶ。

シンクは、文献 [3] と同様の方法によって境界線を推定する。具体的には、収集したセンサデータに含まれる位置情報を母点とするポロノイ図を作成し、観測値が互いに異なる母点のポロノイ境界を境界線として推定する (図 8)。

以上の環境で、端末数  $N$  を変化させ、以下の三つの評価値を調べた。

- **トラヒック**：シンクおよびすべての端末が送信したパ

表 1 メッセージサイズ

| 手順       | メッセージ名  | サイズ [B]          |
|----------|---------|------------------|
| クエリ伝播    | クエリ     | 78               |
| 境界線検出    | データパケット | 14 + 24          |
| センサデータ収集 | 返信パケット  | 16 + 24 <i>i</i> |
|          | ACK     | 14               |

ケットのアプリケーション層でのデータサイズの合計。各メッセージサイズを表 1 に示す。なお、5.2 節の動作による影響を検証するため、境界線検出時におけるトラヒック（各端末がクエリを送信してから転送木上の親へセンサデータを送信するまでに発生したトラヒック）も個別に計測した。

- **平均誤差**：図 7(b) に示す境界線と、シンクが推定した境界線によって囲まれる領域の 1 メートルあたりの面積の平均値。この値は、シンクが推定した境界線と実際の境界線との差異が大きいほど大きくなる。
- **最大誤差**：図 7(b) に示す境界線と、シンクが推定した境界線との距離の最大値。

### 6.2 比較手法

OBD では、5.2 節で述べたとおり、傍受したセンサデータを基に作成したダイアグラムを用いて境界線を検出する。実験では、この動作による効果を検証するために、境界線の検出における処理として、5.2 節で述べた端末間の距離のみを用いる手法を比較対象とし、その性能を OBD と比較した。この手法では、クエリ伝播、センサデータ収集は OBD と同じ手順で行う。境界線検出の処理としては、各端末が自身を中心とした異なる半径  $r_1$ 、 $r_2$  をもつ二つの円内において送信されたセンサデータに基づいて、自身のセンサデータを送信するか否かを判定する。具体的には、観測値が異なるクエリまたはセンサデータを受信した端末は、その送信元端末に近いほど短く設定されるタイマを設け、タイマが切れた時点で以下の条件を満たす場合にセンサデータを送信する。

- (1) 自身を中心とした半径  $r_1$  の円内で、観測値が異なるセンサデータが送信されている。
- (2) 自身を中心とした半径  $r_2$  の円内で、観測値が同じセンサデータが送信されていない。

センサデータを受信した各端末は、センサデータ収集開始時刻までに受信したすべてのセンサデータを基にポロノイ図を作成し、観測値が異なる端末とポロノイ領域上で隣接する場合、転送木上の親端末へ自身のセンサデータを送信する。これにより、境界線に接していると判断した端末のセンサデータのみを収集できる。以降では、この手法を比較 ( $r_1, r_2$ ) と表記する。

パラメータ ( $r_1, r_2$ ) の設定として、まず既存研究のように、境界線が自身の通信範囲内に存在する全端末の情報を取得する方法となる ( $r_1, r_2$ )=(100, 0) を用いた。この手法

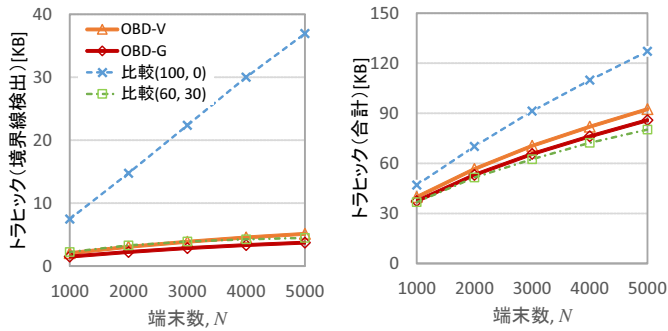


図 9 トラヒック (境界線検出)

図 10 トラヒック (合計)

では、各端末が全隣接端末の情報を取得した後、OBD-V と同様にボロノイ図を作成し、自身が境界線に接していると判断した場合にのみセンサデータを送信する。そのため、パケット衝突がなければ、比較 (100, 0) によりシンクに収集されるセンサデータは、OBD-V を用いた場合とほぼ同じになる。また、端末数  $N$  を 2,000 台とした準備実験を行い、ネットワーク全体のトラヒックが OBD より小さくなり、かつその中で最も平均誤差が小さくなる設定として、 $(r_1, r_2) = (60, 30)$  を用いた。

### 6.3 実験結果

#### 6.3.1 トラヒック

結果を図 9 および図 10 に示す。グラフの横軸は端末数  $N$  を表し、縦軸はそれぞれ、図 9 は境界線検出時のトラヒック、図 10 はトラヒックの合計を表す。

図 9 より、各手法ともに、端末数の増加に伴い、境界線検出時のトラヒックが増加することがわかる。これは、端末数の増加とともに、境界線の近くに存在する端末数も増加するためである。また OBD におけるトラヒックが、比較 (100, 0) に比べて大幅に小さくなることわかる。これは、比較 (100, 0) では、6.2 節で述べた条件 (2) を満たす全端末がセンサデータを送信するのにに対し、OBD ではダイアグラム上で観測値が異なる端末が隣接する端末のみがセンサデータを送信するためである。また、OBD-G におけるトラヒックが OBD-V より小さくなることわかる。これは 5.2.2 項で述べたとおり、ガブリエルグラフを用いたほうが、センサデータを送信する端末数を削減できるためである。また、図 10 からわかるとおり、ネットワーク全体のトラヒックも、比較 (100, 0) と比べて OBD が小さくなる。一方、比較 (60, 30) は、準備実験に基づくパラメータ設定により、全体のトラヒックが OBD より小さくなる。

図 9 におけるトラヒックの差異を詳細に検証するため、端末数が 2,000 台のときに、各手法によりセンサデータを送信した端末 (送信端末) の 2 次元平面上における分布を調査した。結果を図 11 に示す。図では、各手法における送信端末数も示している。図 11(a) および図 11(b) より、OBD によりセンサデータを送信した端末は、境界線付近に

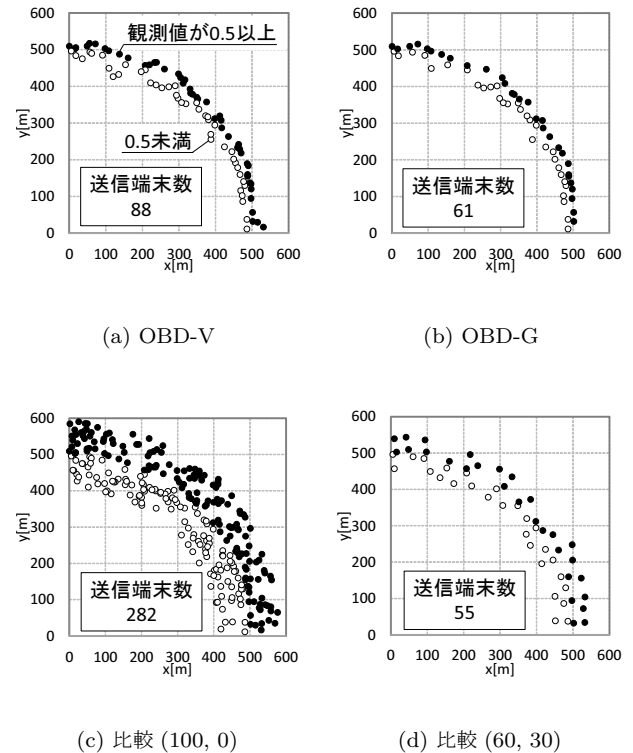


図 11 センサデータを送信した端末の分布

位置する端末のみであることがわかる。また OBD-G は、OBD-V より送信端末数が少ないことがわかる。一方、図 11(c) より、比較 (100, 0) を用いた場合、非常に多くの端末がセンサデータを送信していることがわかる。また、図 11(d) より、比較 (60, 30) によりセンサデータを送信した端末数は、OBD より少なくなることがわかる。しかし、OBD と異なり、多くの境界線付近の端末もセンサデータを送信していないことがわかる。

#### 6.3.2 平均誤差および最大誤差

結果を図 12 および図 13 に示す。グラフの横軸は端末数  $N$  を表し、縦軸はそれぞれ、図 12 は平均誤差、図 13 は最大誤差を表す。

図 12 より、各手法ともに、端末数の増加に伴い平均誤差が小さくなることわかる。これは、端末数が増加すると境界線を推定する際に使用するセンサデータ数が増えるためである。また、比較手法では、トラヒックが小さくなる比較 (60, 30) における平均誤差が、比較 (100, 0) より大きくなっており、 $(r_1, r_2)$  の設定によってトラヒックと平均誤差の間にトレードオフが発生することがわかる。5.2 節で述べたとおり、比較手法は  $(r_1, r_2)$  の設定によってトラヒックを削減できるが、 $r_1$  を小さく設定し過ぎると、観測値が異なるクエリの転送元端末との距離が  $r_1$  以上になる可能性が高くなり、境界線の検出が開始されない。また  $r_2$  を大きく設定し過ぎると、境界線に十分近い端末までセンサデータを送信しない。この結果から比較手法では、トラ



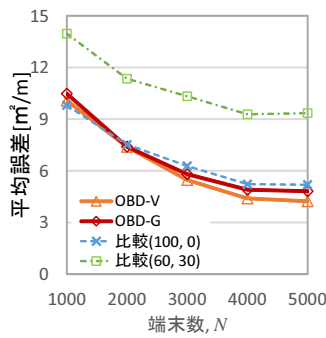


図 12 平均誤差

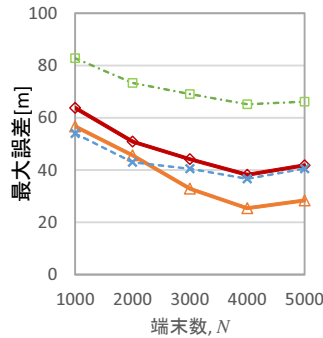


図 13 最大誤差

ヒックを抑えながら平均誤差が小さい境界線を推定できる ( $r_1, r_2$ ) の設定が困難であることがわかる。

一方、OBD における平均誤差は比較 (100, 0) と同等またはそれ以下の値を示しており、トラヒックを削減しつつ、平均誤差が小さい境界線を推定可能であることがわかる。ここで、OBD-G における平均誤差が、OBD-V と比較して若干大きくなっている。これは前項でも述べたとおり、OBD-G では、OBD-V よりセンサデータを送信する端末数が少なく、結果としてシンクが境界線を推定する際に使用するセンサデータ数が、OBD-V より少なくなるためである。また端末数の増加に伴い、OBD-V と比較 (100, 0) の平均誤差の差が大きくなることがわかる。6.2 節で述べたように、比較 (100, 0) によりシンクに収集されるセンサデータは、パケット衝突がなければ、OBD-V を用いた場合とほぼ同じになる。しかし、比較 (100, 0) では、条件 (2) を満たす全端末がセンサデータを送信するため、端末数の増加に伴い、センサデータ送信時にパケット衝突が発生しやすくなる。そのため、比較 (100, 0) における一部の端末が OBD-V と同じボロノイ図を作成できず、実際は境界線に接しない端末がセンサデータを送信してしまう。これらの端末が送信したセンサデータが境界線の推定に使用されると、誤差が大きくなる領域が現れる。

また、図 13 より、OBD-G における最大誤差は、比較 (100, 0) よりも常に大きくなることがわかる。これは、6.1 節で述べたシンクにおける境界線の推定方法によるものである。本実験では、文献 [3] と同様に、ボロノイ境界を用いて境界線を推定しているが、OBD-G を用いた場合、母点となる端末の一部がセンサデータを送信せず、誤差が大きくなる領域が現れる。このような誤差の増大を抑制するためには、シンクにおいて、特定のダイアグラムに依存しない境界線の推定方法を用いる必要があるものと考えられる。

## 7. まとめ

本稿では、端末が密に存在するモバイルセンサネットワークにおいて、各端末がセンサデータの傍受を用いて効率的に観測値の境界線を検出する手法である OBD を提案した。OBD では、傍受したセンサデータを基に各端末が

ボロノイ図またはガブリエルグラフを作成し、自身が境界線に接しているかどうかを判断する。また、センサデータを送信するまでのタイマを設定することにより、境界線に接している端末のみがセンサデータを送信し、境界線検出時のトラヒックを削減できる。シミュレーション実験により、OBD は境界線検出時のトラヒックを削減しながら、高い精度で境界線を推定できることを確認した。

本稿では、シンクがクエリを発行した時点における境界線の検出を想定しているため、文献 [4] で想定しているような境界線の移動は考慮していない。しかし、ユーザが境界線の定期的なモニタリングを要求する場合、モニタリング中に境界線が移動するものと考えられる。そこで今後は、時間経過に伴い移動する境界線を効率的にモニタリングする方法について検討する予定である。

**謝辞** 本研究の一部は、文部科学省科学研究費補助金・基盤研究 (S)(21220002)、基盤研究 (B)(24300037)、および日本学術振興会科学研究費補助金 (B)(23700078) の研究助成によるものである。ここに記して謝意を表す。

## 参考文献

- [1] Camp, T., Boleng, J. and Davies, V.: A survey of mobility models for ad hoc network research, *Wireless Communications and Mobile Computing*, Vol.2, No.5, pp.483–502 (2002).
- [2] Estrin, D. L.: Participatory sensing: applications and architecture, *Proc. MobiSys 2010*, pp.12–42 (2010).
- [3] Ham, M. I. and Rodriguez, M. A.: A boundary approximation algorithm for distributed sensor networks, *Int. Journal of Sensor Networks*, Vol.8, No.1, pp.41–46 (2010).
- [4] Jiang, J., Worboys, M. and Nittel, S.: Qualitative change detection using sensor networks based on connectivity information, *GeoInformatica*, Vol.15, No.2, pp.305–328 (2011).
- [5] Li, M. and Liu, Y.: Iso-map: energy-efficient contour mapping in wireless sensor networks, *IEEE Trans. on Knowledge and Data Engineering*, Vol.22, No.5, pp.699–710 (2010).
- [6] Tseng, Y., Ni, S., Chen, Y. and Sheu, J.: The broadcast storm problem in a mobile ad hoc network, *Wireless networks*, Vol.8, No.2–3, pp.153–167 (2002).
- [7] Scenargie1.6 Base Simulator revision 11773, Space-Time Engineering, <https://www.spacetime-eng.com/>
- [8] Solis, I. and Obraczka, K.: Isolines: efficient spatio-temporal data aggregation in sensor Networks, *Wireless Communications and Mobile Computing*, Vol.9, No.3, pp.357–367 (2009).
- [9] Srinivasan, S., Dattagupta, S., Kulkarni, P. and Ramamritham, K.: A survey of sensory data boundary estimation, covering and tracking techniques using collaborating sensors, *Pervasive and Mobile Computing*, Vol.8, No.3, pp.358–375 (2012).
- [10] Tuncay, G. S., Benincasa, G. and Helmy, A.: Autonomous and distributed recruitment and data collection framework for opportunistic sensing, *ACM SIGMOBILE Mobile Computing and Communications Review*, Vol.16, No.4, pp.50–53 (2013).