

密なモバイルセンサネットワークにおける境界線の形状を考慮したサンプリングを用いた境界線検出手法

松尾 和哉¹ 後藤 啓介¹ 神崎 映光² 原 隆浩¹

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、観測値が一定値以上となる性質をもつイベントの形状を、指定した時刻までに取得することをアプリケーションが要求する場合、イベントの地理的境界(境界線)付近に位置する端末を識別し、それらの端末が取得した観測値および位置情報を、指定した時刻までに収集して境界線の形状を推定することで、この要求を満足できる。上記は、筆者らがこれまでに提案している、境界線付近の端末を効率的に識別する手法を用いることによって実現できる。しかし、指定した時刻までの時間が非常に短い場合、この手法が識別した端末のデータを全て収集しようとする、短期間に集中して多くの通信が発生し、パケット衝突の頻発によって多くのデータ収集に失敗してしまう。また、結果的に収集に失敗するデータのためのトラヒックが大量に発生し、無線通信帯域を無駄に消費してしまう。そこで本稿では、筆者らが提案している手法によって識別される端末を、境界線の形状に応じて地理的にサンプリングすることで、境界線の推定精度を維持しつつ、収集するデータ数を削減する手法を提案する。さらに本稿では、提案手法の有効性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、半導体技術の発展により、スマートフォンやウェアラブルデバイスなどの人々が身につけて持ち歩くデバイスに、音や温度、湿度などの環境情報を取得できるセンサが搭載されつつある。これらの移動型センサ端末の普及に伴い、一般ユーザがもつ端末が取得したデータ(環境情報および位置情報)を提供してもらい、都市部における環境モニタリングなどのサービスに活用するユーザ参加型センシングが注目されている[4]。これを実現する方法の一つとして、Wi-Fi DirectやBluetoothなどによる移動型センサ端末間の直接通信により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[7]。モバイルセンサネットワークでは、端末は自身が取得したデータを、端末間の無線マルチホップ通信を用いて、データを収集する端末であるシンクへ送信する。

ここで、繁華街などの人通りの多い場所で構築された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、各端末に対して、直接通信可能な端末(隣接端末)が常に多数存在する。本稿では、これを密なモバイルセンサネット

ワークと呼ぶ。このような環境において、すべての端末からデータを収集すると、そのためのトラヒックが膨大となり、有限な通信帯域を圧迫してしまう可能性がある。そのため、アプリケーションの要求に応じて必要最小限のデータのみを収集することが求められる。ここで、モバイルセンサネットワークを用いるアプリケーションには、「局所的に気温が高い場所を特定したい」など、観測値が一定値以上を示す領域(イベント)の形状を要求するものが考えられる。この要求を満足するためには、イベントの地理的境界(境界線)が取得できればよい。このような境界線は、その境界線付近に位置する端末のデータのみを収集し、イベントを観測しているデータとイベントを観測していないデータの間を線を引くことで、十分に推定できる[1], [3]。このような背景のもと、筆者らは文献[5]において、データの傍受とダイアグラムを用いて境界線付近の端末を効率的に識別する手法を提案している(以降、従来手法と呼ぶ)。この手法では、各端末が、他の端末から傍受したデータから端末の位置に基づくダイアグラムを作成し、これを基に自身が境界線付近に位置すると判断した場合にのみ、シンクへデータを送信する。本稿では、境界線付近の端末を識別し、それらの端末からデータを収集し、境界線を推定する一連の処理を、「境界線検出」と表現する。

¹ 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

² 島根大学大学院総合理工学研究科総合理工学専攻

モバイルセンサネットワークにおいて要求されることが想定されるイベントの例として、上述した気温が高い場所の他に、局所的に降雨量が多くなるゲリラ豪雨や、PM2.5や花粉などの微粒子の濃度が高い領域なども挙げられる。これらを対象とした場合、毎秒数十メートル以上の速さでその位置や形状が変化することが考えられるため、アプリケーションが要求した時刻の境界線が大きく変化してしまう前に、数百ミリ秒程度のごく短時間でデータ収集を完了することが重要である。このような環境では、アプリケーションは境界線検出のために許容できる最大遅延に基づき、データの収集が完了すべき時刻（収集完了時刻）を設定するものと考えられる。しかし、上述した従来手法は、収集完了時刻が設定される要求を想定しておらず、この時刻までにデータ収集を完了することは困難である。これは、従来手法において、境界線付近に位置すると判断されたすべての端末からデータを収集することに起因する。従来手法では、特に収集するデータ数が多くなった場合を考慮して、パケット衝突が発生しないよう、十分な待ち時間を設定したデータ収集を行う。このデータ収集を非常に短い時間内に完了させようとする、短期間に集中して多くの通信が発生し、パケット衝突の頻発によって多くのデータの収集に失敗してしまう。また、結果的に収集に失敗するデータのためのトラフィックが大量に発生、無線通信帯域を無駄に消費してしまう。

この問題を解決するためには、できる限り高い精度での境界線推定を実現しつつ、収集するデータ数を短時間に収集可能な数まで減らす必要がある。これを実現するために、本稿では、境界線付近の端末を地理的にサンプリングする手法を提案する。提案手法では、まず従来手法が識別する端末を、互いの距離が閾値以上となるようにサンプリングし、これらの端末のみがデータをシンクへ送信する。また、このサンプリングによる境界線の推定精度低下を抑えるため、送信されたデータを傍受した他の端末が、傍受したデータを用いて、自身の周辺における境界線を部分的に推定し、これを基に、推定精度の向上が見込める場合のみ、シンクへ収集するデータの追加や削除を行う。これにより、境界線の形状に応じて、より適切なサンプリングが行われるため、推定する境界線の精度を維持しつつ、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る。3章で従来手法について述べ、4章で提案手法について述べた後、5章でシミュレーション実験の結果を示す。最後に6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

2.1 端末構成

本研究では、一般ユーザの携帯する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、後述するシンクの位置は固定されているものとする。

2.2 アプリケーション要求

観測対象となる領域（観測領域）は2次元平面を想定する。アプリケーションは観測値の閾値を設定し、観測値が閾値以上の値を示すイベントの境界線を取得することを要求する。シンクは、データ収集のために必要な情報を含むクエリを発行し、観測領域内の端末からデータ（観測値とセンシング時の位置情報）を収集する。またシンクは、収集したデータを用いて、次節で述べる方法で境界線を推定する。

2.3 境界線推定方法

シンクにおける境界線の推定は、ドロネー図と呼ばれるダイアグラムを、端末の位置を基に生成して行うものとする。ドロネー図とは、ある3点を通る同一円内に他の点が存在しないという条件を満たす3点を、互いに辺で結んだ三角形（ドロネー三角形）で構成される図である。ドロネー三角形の各頂点に該当する点は、他の頂点よりも互いに近傍に存在するという特性をもつため、この特性を利用して、データが観測された位置を頂点としたドロネー三角形を生成し、これに基づいて、境界線を推定する。具体的には、データが観測された位置を頂点とする三角形のうち、その外接円内に他のデータが観測された位置が存在しないものを選出する。これらの三角形の辺の中で、イベント内に位置する点と、イベント外に位置する点を結ぶものの中点どうしを結ぶ線分の集合を境界線とする（図1）。便宜上、以降では、イベント内の観測値とイベント外の観測値を、互いに異なる観測値と表現する。

3. 従来手法

本章では、筆者らが文献[5]で提案した手法について述べる。従来手法では、3.1節で述べる方法で、クエリを観測領域全体に伝播した後、3.2節で述べる方法で、境界線付近の端末を識別する。その後、3.3節で述べる方法で、データを収集する。

3.1 クエリ伝播

2.1節で述べたとおり、各端末はデータ収集に関する情報をもっていないため、データ収集の際には、観測領域全

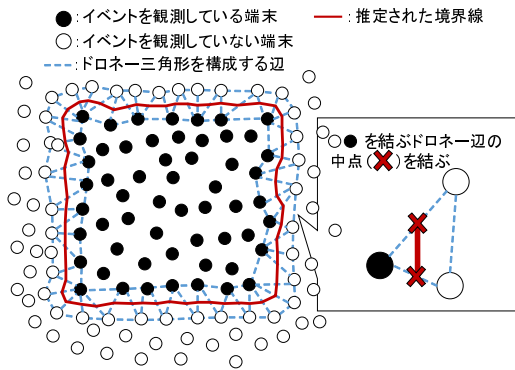


図 1 境界線の推定方法

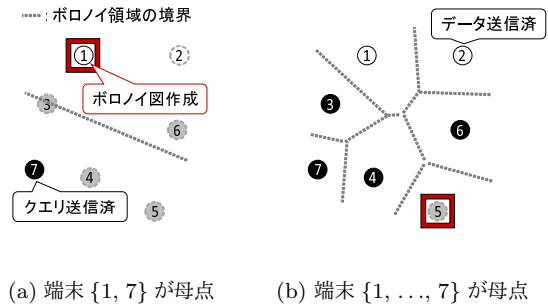


図 2 境界端末の識別の例

体にクエリを伝播する必要がある。

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、アプリケーションの要求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、転送元端末の識別子、位置、観測値を含み、後述する境界線付近の端末の識別処理の終了条件として、データ収集開始時刻 t_{gather} 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、自身の通信範囲内の端末へ転送する。クエリを受信した各端末は、自身の位置におけるイベントの有無の観測、および位置情報の取得を行う。その後、クエリ転送のための待ち時間を、受信したクエリの転送元端末から遠いほど短くなるように設定する。待ち時間内に他の端末が転送したクエリを傍受し、他のクエリ転送元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、クエリ内の転送元端末に関する情報を自身のものに更新し、クエリを転送する。このとき、クエリを転送する端末が自身の観測値をクエリに付与することで、異なる観測値が含まれたクエリを受信した端末が、自身の通信範囲内に境界線があることを把握できる。また各端末は、自身にクエリを転送した端末の識別子を親端末の識別子として記録し、この情報を用いてシンクを根、クエリを転送した各端末を節点とする転送木を構築する。この転送木は、3.3 節で述べるデータ収集時に利用する。

3.2 境界線付近の端末の識別

この処理では、ドロネー図と双対の関係にあるポロノイ図を用いて境界線付近の端末を識別する。ポロノイ図とは、平面上の任意の点を最も近い母点に属する領域（ポロノイ領域）として分割した図であり、互いのポロノイ領域が隣接する（ポロノイ領域上で隣接する）二つの母点を結んだ線分がドロネー辺となる。ポロノイ領域上で隣接する

母点は、他の母点より近傍に存在するため、この特性を利用することで、各端末を母点とするポロノイ図において、ポ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と判断できる。以降では、対象領域の全端末を母点とするポロノイ領域上で隣接する端末の観測値が異なる場合、「境界線に接する」と表現し、境界線に接する端末を「境界端末」と表現する。

従来手法では、本節の処理において、境界端末であると判断した端末のみがデータを送信することにより、境界端末識別に要するトラヒックを最小限に抑えている。このときの各端末の動作を、図 2 を例に説明する。この図は、前節の処理によって端末 7 がクエリを送信しており、これを端末 {1, ..., 6} が受信した場合を示している。色が薄い端末はデータを送信しておらず、他の端末から認識されていないことを示す。また、これらの端末は互いに直接通信可能であり、互いが送信したデータを受信可能であるものとする。

まず、各端末は、前節の処理において転送されたクエリ、または本節の処理において送信されたデータを傍受するたびに、それらのデータを用いてポロノイ図を作成する。このポロノイ図を基に、データ収集開始時刻 t_{gather} になるまで、以下のとおり動作する。

- ポロノイ領域上で隣接する端末の観測値が異なる場合、その端末との距離が小さいほど短くなるタイマを設定する。タイマが経過した後に、自身のデータを 3.1 節で構築した転送木上の親端末へ送信し、本節の処理を終了する。図 2(a) では、クエリを受信した端末 1 が、自身および端末 7 の位置を母点とするポロノイ図を作成している。図からわかる通り、端末 1 は、自身と観測値が異なる端末 7 とポロノイ領域上で隣接するため、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。その結果、より端末 7 に近い端末 1 のタイマが先に切れ、転送木上の親である端末 7 へデータを送信する。
- ポロノイ領域上で観測値が異なる端末と隣接していない、あるいは隣接しなくなった場合、データは送信しない。例えば図 2(b) は、端末 {1, 2, 3, 4, 6} が端末 7

ヘータを送信した後、これらの端末のデータを傍受した端末5が作成したボロノイ図である。図からわかる通り、端末5は、観測値が異なる端末{1, 2}のデータを傍受しているものの、ボロノイ領域上で隣接する端末は{4, 6}のみであり、この中に観測値が異なる端末が存在しないため、データを送信しない。

3.3 データ収集

データ収集開始時刻になった時点で、3.1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、パケットを送信した子端末へACKを送信する。ACKを受信しなかった子端末は、親端末のACKが受信できるまで一定回数メッセージを再送する。自身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、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する。この待ち時間に従ってパケットを送信することで、シンクから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データが、シンクに収集される。

3.4 従来手法の問題点

従来手法では、収集完了時刻が指定されることを想定しておらず、3.2節で送信されるデータを、パケット衝突を避けながらすべて収集するため、収集用の待ち時間を十分に長く設定する必要がある。収集完了時刻までのデータ収集を実現しようとした場合、この待ち時間を短く設定することが考えられるが、この場合、短期間にパケットの送信が集中し、パケット衝突および再送が頻発する。結果として、シンクが複数のパケットを収集完了時刻までに受信できなくなる。特に従来手法では、3.3節で述べた処理によって、複数のデータをひとつのパケットに集約して収集するため、パケットの収集失敗によって多数のデータが推定に使用できなくなる。また、1パケットあたりのサイズが大きくなるため、パケットの再送が頻発した場合、トラヒックが極端に増大してしまう。その結果、最終的に収集に失敗するデータのためのトラヒックが大量に発生し、通信帯域を無駄に消費してしまう。

4. 提案手法

本章では、まず、提案手法が想定するアプリケーション要求について述べた後、提案手法の設計方針を述べる。次に提案手法の概要を述べ、その後、提案手法の詳細な手順について説明する。

なお、以降では、観測領域内の任意の位置における観測値が把握できたときのイベントの境界線を、説明の便宜上、

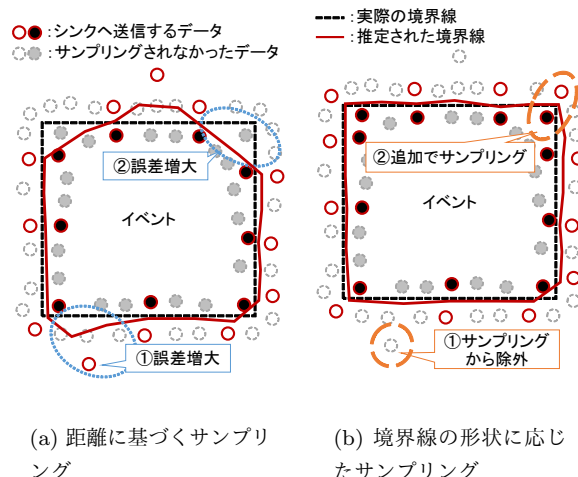


図3 提案手法によるサンプリング

実際の境界線と表現する。また、実際の境界線から、シンクが推定した境界線までの距離を、推定精度を示す指標として、「推定誤差」と表現する。

4.1 アプリケーション要求

基本的には2.2節で述べたものと同様であるが、これに加え、アプリケーションは収集完了時刻 t_{end} を設定し、この時刻までにデータを取得することを要求する。

4.2 設計方針

3.4節で述べたとおり、収集完了時刻までの時間が非常に短く、かつ3.2節でデータを送信する端末が多い場合、短時間に集中して通信が発生し、パケット衝突の頻発によって多くのデータ収集に失敗する。結果として、境界線の推定に利用できるデータ数が極端に少なくなり、推定誤差が増大する。また、結果的に収集に失敗するデータのためのトラヒックが大量に発生しており、通信帯域を無駄に消費してしまう。

1章で述べたとおり、この問題を解決するためには、推定誤差をできる限り低く保ちつつ、収集するデータ数を短時間に収集可能な数まで減らす必要がある。そのための単純な方法として、図3(a)のように、従来手法が識別する境界端末の中から、データを送信する端末を、観測値が同じ端末間の距離が閾値以上となるようにサンプリングする方法が考えられる。しかし、単純に端末間の距離のみでサンプリングを行うと、図3(a)の点線で囲んだ領域のように、推定誤差が大きくなってしまう可能性がある。そこで、提案手法では、端末間の距離のみを考慮した境界端末のサンプリングに加え、図3(b)に示すように、各端末が、境界線の形状に応じて、自身のデータを送信する、または送信を中止する処理を行う。これにより、推定誤差の増大を抑えつつ、収集するデータ数を削減できる。

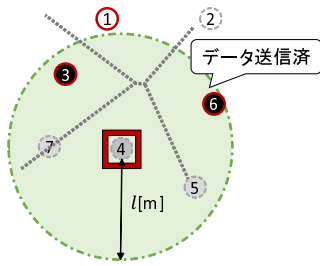


図 4 距離に基づくサンプリングの例

4.3 動作概要

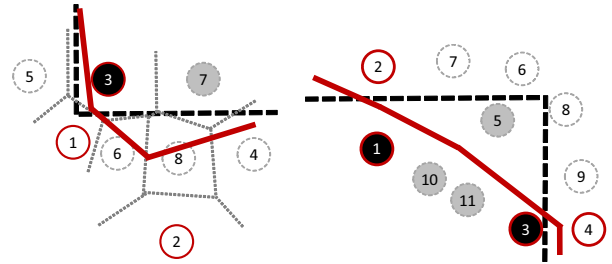
提案手法では、従来手法と同様に、3.1節で述べる方法で、クエリを観測領域全体に伝播する。ここで、クエリには、3.1節で述べた情報に加え、収集完了時刻 t_{end} を付与する。クエリを受信した端末は、4.4節および4.5節で述べる方法で、境界線の形状に応じて、境界端末を識別しつつ、収集するデータ数を削減する。その後、4.6節で述べる方法で、データを収集する。

4.4 距離に基づく地理的なサンプリング

基本的な動作は3.2節と同様であるが、データを送信する端末間の距離が閾値 l 以上となるように、境界端末をサンプリングする。具体的には、ポロノイ領域上で観測値の異なる端末と隣接しても、自身を中心とする半径 l ($\leq r$) の円内に観測値が同じデータを送信済みである端末が存在する場合、データの送信を中止する。このときの動作例を図4に示す。この図は、図2において、端末 {1, 3, 6} の順にデータが送信済みである状態を示している。これらのデータを傍受した端末4は、観測値が異なる端末1とポロノイ領域上で隣接しているが、自身を中心とする半径 l の円内に位置し、自身と観測値が同じ端末 {3, 6} がデータを送信済みであるため、データの送信を中止する。

以上の手順によって、観測値が同じ端末が閾値 l 以上の間隔でサンプリングされ、送信されるデータ数を削減できるが、以下の二つの問題が発生する。

- (1) 図5(a)において、従来手法では識別される端末6や8がデータの送信を中止することによって、端末2のように、実際には境界線と接さない端末が、自身が境界端末であると誤って判断し、サンプリングされる可能性がある。実際の境界線に対して、境界端末より遠くに位置する端末がデータを送信した場合、推定される境界線が、データを送信した端末が位置している方向へずれてしまい、推定誤差が増大する可能性がある。また、端末4のように、実際に境界線と接する端末のデータ送信を妨げてしまう可能性もある。
- (2) 図5(b)における端末5のように、推定された境界線による観測値の分類が誤っている位置にいる境界端末がサンプリングされない可能性がある。この端末のデータなしで境界線を推定した場合、実際の境界線から離



(a) 誤って境界端末としてサンプリングされる。

(b) 推定された境界線による観測値の分類が誤っている位置にいる境界端末がサンプリングされない。

図 5 距離に基づくサンプリングによって推定誤差が増大する例

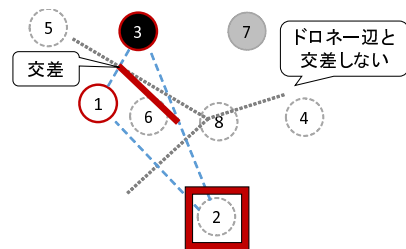


図 6 サンプリング対象からの除外の例

れた境界線を推定してしまう。

4.5 境界線の形状に応じたサンプリング

前節で述べた問題を解決するために、各端末は、自身がデータを送信する、またはデータ収集開始時刻 t_{gather} になるまで、前節の処理で送信されたデータから、シンクと同じ手順によって部分的に境界線を推定し、その境界線を用いて、サンプリングの追加と除外を行う。

4.5.1 サンプリング対象からの除外

4.4節で述べた問題点(1)を解決するためには、各端末が、自身が実際に境界線と接する可能性があるかどうかを、自身が傍受したデータのみから判断しなければならない。ここで、ポロノイ図の特性上、ポロノイ領域の境界はドローネー辺の垂直二等分線であり、図6における端末1と3のように、互いに近傍に位置する端末間のドローネー辺は、当該端末を母点とするポロノイ領域の境界と交差しやすくなる。一方、図6における端末2と3のように、他の端末と比べて端末間の距離が大きくなると、ドローネー辺とポロノイ領域の境界が交差しなくなる。本稿ではドローネー辺の中間点を結ぶ線分の集合として境界線を推定するため、ドローネー辺とポロノイ領域の境界が交差しない場合、端末2のように、推測される部分的な境界線が、ポロノイ領域と接さない状態となる。これは、他の端末と比較して、推測される部分的な境界線と当該端末との距離が大きく、境界線に接していない可能性があることを意味する。提案手法で

は、この特性を利用し、推定誤差増大の要因となる可能性がある端末のデータ送信を抑制する。具体的には、4.4節の処理により、データ送信のためのタイマを設定している境界端末は、設定したタイマが切れる前に、傍受したデータと自身のデータを用いて、2.3節と同様の手順により部分的に境界線を推定する。この境界線が、自身のボロノイ領域と接さない場合、自身は実際には境界線と接さない可能性があると判断し、データの送信を中止する。

4.5.2 追加のサンプリング

4.4節で述べた問題点(2)を解決するために、前節の処理によってサンプリング対象にならなかった端末は、傍受したデータを用いて、境界線を部分的に推定する。例えば図7において、端末{1, ..., 4}のデータを傍受した端末5は、これらのデータを用いて部分的に境界線を推定する。このとき、端末5自身はサンプリング対象となっていないため、境界線の推定には用いない。その後、この処理によって推定された境界線による、自身の位置における観測値の分類が正しくない場合、推定される境界線を修正するためのデータ(修正用データ)を送信するためのタイマを、推定した境界線からの距離が大きいほど短くなるように設定する。このタイマが切れるまでに、自身を中心とする半径 l の円内に位置する端末から、観測値が同じ修正用データを傍受しなかった場合のみ、修正用データであることを示すフラグを付与したデータを親端末へ送信し、本節の全サンプリング処理を終了する。例えば図7の端末5は、部分的に推定した境界線による、自身の位置における観測値の分類は「イベントなし」であり、実際の観測値と異なっているため、この境界線から速いほど短くなるタイマを設定し、修正用データを送信する。

また、上記の処理で送信された修正用データを傍受した端末のうち、送信元端末と観測値が異なる境界端末は、自身を中心とする半径 l の範囲内に自身と観測値が同じ端末が存在する場合においても、送信元端末に近いほど短くなるタイマを設定する。タイマが切れるまでに、自身を中心とする半径 l の円内に位置する端末のうち、データを送信していない端末から、観測値が同じデータを新たに傍受しなかった場合のみ、自身のデータを親端末へ送信し、本節のサンプリング処理を終了する。例えば図7において、端末5が送信した修正用データを傍受した端末6は、端末5に近いほど短くなるタイマを設定し、自身のデータを送信する。

4.6 データ収集

境界端末の識別およびサンプリング処理において、最も遅く送信されるデータは、データ送信開始時刻 t_{gather} に送信される。このデータを収集完了時刻 t_{end} までにシンクが受信するためには、このデータの収集を $(t_{end} - t_{gather})$ の期間内に行う必要がある。これを実現するため、以下の

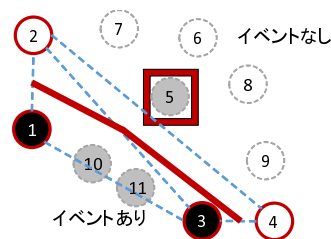


図7 追加のサンプリングの例

式に従って、受信したデータを自身の転送木上の親端末へ送信する時間を設定する。

$$W^g = \frac{t_{end} - t_{gather}}{h_{max}} - trans-delay + rand. \quad (1)$$

ここで、 h_{max} はシンクからのホップ数の最大値、 $trans-delay$ は1ホップあたりの転送遅延、 $rand$ はパケット衝突を回避するために与える乱数を示す。上式の第1項は、 $(t_{end} - t_{gather})$ の期間内に1ホップあたりのデータ送受信に使用できる時間であり、この時間から1ホップあたりの転送遅延を引くことで、自身がデータを受信してから、そのデータを送信するまでに待つことができる時間を算出できる。転送木上の節점에該当する端末は、サンプリングされたデータを受信するたびに、受信してから W^g 経過後、受信したデータを自身の転送木上の親端末へ送信する。また、再送の頻発によるトラヒックの増大を防ぐため、データの再送は行わないものとする。上記の処理により、データ収集開始時刻 t_{gather} までに送信された境界端末のデータを、パケット衝突が発生しない限り、シンクは収集完了時刻までに受信できる。

5. 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、ゲリラ豪雨や、PM2.5や花粉などの微粒子の濃度が高い領域など、風に乗って移動するイベントの境界線を検出する環境を想定した。実験には、ネットワークシミュレータ Scenargie 2.0^{*1}を用いた。

5.1 シミュレーション環境

1,000[m]×1,000[m]の2次元平面上に、1,000台の移動型センサ端末および1台のシンクが存在する。シンクは、観測領域の左下端に配置した。各端末はランダムウェイポイントモデル[2]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、 $r = 100$ [m]程度となる送信電力でパケットを送信する。

シンクは、シミュレーション開始から収集完了時刻 t_{end}

^{*1} Scenargie2.0 Base Simulator revision 18021, Space-Time Engineering, November 2015. [Online]. Available: <https://www.spacetime-eng.com/>

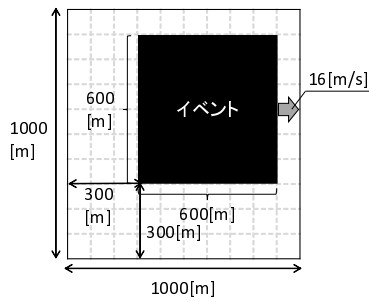


図 8 シミュレーション実験で用いるイベント

表 1 メッセージサイズ

Table 1 Message size.

手法	手順	メッセージ名	サイズ [B]
提案	境界端末の識別 およびサンプリング	データパケット	21 + 24
従来および 単純	境界端末の識別 およびサンプリング	データパケット	20 + 24
共通	クエリ伝播 データ収集	クエリ 返信パケット	92 26 + 24

($0.4[s] \leq t_{end} \leq 1[s]$) までデータ収集を行うものとする。データ収集開始時刻は、各観測時刻から $0.3 (t_{gather} = 0.3)$ [s] 後とした。

イベントの形状は、一辺が $600[m]$ の正方形とした。このイベントは図 8 の位置を初期位置とし、矢印の方向に $16[m/s]$ の速度で等速移動するものとする。

5.2 評価指標

アプリケーションが収集完了時刻を設定する場合、収集完了時刻の値によって、データ収集に使用できる時間が変わるため、パケット衝突の回数が増えるものと考えられる。そこで本実験では、上記の環境において、収集完了時刻 t_{end} を変化させ、以下の三つの評価値を調べた。

- **ロス率**：境界端末の識別時に送信された全データ数に対する、シンクが受信できなかったものの割合。
- **トラヒック**：シンクおよびすべての端末が送信したメッセージのアプリケーション層におけるサイズの総計。各メッセージのサイズを表 1 に示す。
- **推定誤差**：実際の境界線と、シンクが推定した境界線によって囲まれる領域を、実際の境界線の長さで割った値。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。また、各手法を用いて収集したデータから推定した境界線の有効性を調べるために、クエリ発行時刻における観測領域内の全端末のデータを用いて推定した境界線の推定誤差を、理想値として個別に計測した。

5.3 評価手法

提案手法では、端末間の距離に基づいて境界端末を地理

的にサンプリングし、さらに境界線の形状に応じたサンプリングの追加と除外を行うことで、境界線の推定誤差の抑制とトラヒックの削減を図っている。実験では、この動作による効果を検証するために、評価手法として、以下の三つの手法の性能を調べた。

- **提案手法**：4章で述べた手法に従い、境界線の形状を考慮してサンプリングされた境界端末のデータを収集する手法。4.4節で述べた閾値 l は、 $50[m]$ に設定した。
- **従来手法**：3章で述べた手法に従い、全境界端末のデータを収集する手法。ただし、収集完了時刻までにデータ収集を完了させるため、データ収集は 4.6 節の手順に従って行うものとした。
- **距離ベースサンプリング**：上記の従来手法に、4.4節で述べた処理のみを適用した手法。4.4節で述べた閾値 l は、準備実験を行い、この手法のトラヒックが提案手法とほぼ同等になる設定として、 $45[m]$ を用いた。これは、この手法を用いて、提案手法と同程度の数のデータを収集する場合の推定誤差を確認するためである。

5.4 評価結果

結果を図 9、図 10 および図 11 に示す。グラフの横軸は収集完了時刻 t_{end} を表し、縦軸はそれぞれ、図 9 はロス率、図 10 はトラヒック、図 11 は推定誤差を表す。

図 9 より、収集完了時刻までの時間が長くなるほど、各手法におけるロス率が低下していることがわかる。これは、データ収集に使用できる時間が長くなり、パケット衝突の回数が減るためである。また、提案手法および距離ベースサンプリングにおけるロス率が、従来手法より小さいことがわかる。これは、従来手法によって識別される境界端末を地理的にサンプリングすることで、収集データ数を削減しているためである。

図 10 より、収集完了時刻までの時間が長くなるほど、各手法におけるトラヒックが増加していることがわかる。これは、収集完了時刻までの時間が長くなるほど、パケット衝突の回数が減り、より多くのデータがシンクへ送信されるためである。また、提案手法および距離ベースサンプリングにおけるトラヒックが、従来手法より小さいことがわかる。これは、従来手法に対して、サンプリングによって収集されるデータ数を削減しているためである。一方、距離ベースサンプリングは、準備実験に基づくパラメータ設定により、トラヒックが提案手法とほぼ同じになっている。

図 11 より、収集完了時刻までの時間が長くなるほど、各手法における推定誤差が低下し、理想値に近づいていることがわかる。これは、ロス率の低下に伴い、境界線推定時に使用できるデータ数が増えるためである。また、距離ベースサンプリングにおける推定誤差が、従来手法より大きくなることがわかる。これは、単純な距離ベースサンブ

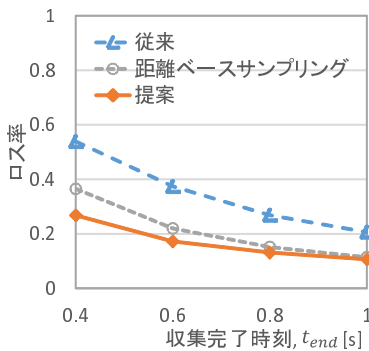


図 9 ロス率

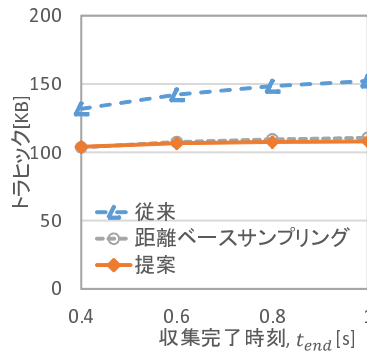


図 10 トラフィック

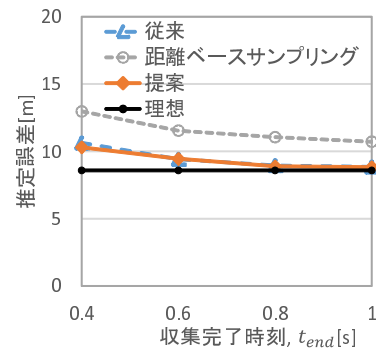


図 11 推定誤差

リングによって、境界線の形状を考慮せずにデータを送信する端末がサンプリングされるため、4.4 節で述べた問題が発生し、誤差が増大するためである。これに対して提案手法は、境界線の形状に応じて、サンプリングする端末を除外、または追加することで、サンプリングによる誤差の増大を抑制しているため、従来手法とほぼ同等の推定誤差を実現している。

以上の結果より、提案手法は、推定誤差の増大を抑えつつ、データ収集のためのトラフィックを削減し、無線通信帯域の無駄な消費を防げることが確認できた。

6. まとめ

本稿は、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、境界線の形状に応じて境界線付近の端末を地理的にサンプリングし、収集するデータ数を削減することで、境界線の推定精度を維持しつつ、トラフィックを削減する手法を提案した。提案手法では、まず従来手法が識別する端末を、互いの距離が閾値以上となるようにサンプリングし、これらの端末のみがデータをシンクへ送信する。このとき送信されたデータを傍受した他の端末が、傍受したデータを用いて、自身の周辺における境界線を部分的に推定する。推定した境界線を基に、精度低下の原因になるデータを所持していると判断した端末は、データの送信を中止し、精度向上に必要なデータを所持すると判断した端末は、シンクへデータを送信する。これにより、境界線の形状に応じたサンプリングが行われるため、推定する境界線の精度を維持しつつ、トラフィックを削減できる。シミュレーション実験により、提案手法は、推定する境界線の推定誤差の増大を抑制しつつ、トラフィックを削減することで、短期間のデータ収集を効率的に実現できることを確認した。

また本稿では、シンクがクエリを発行した時点における境界線を低遅延で取得することを想定しており、境界線のモニタリングは考慮していない。しかし、ユーザが境界線のリアルタイム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変化するものと考えられる。一方、この場合、境界線の情報を短期間でシンクへ送信することが重

要である。そこで今後は、提案手法を基に、時間経過にともない変化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・特別研究員奨励費 (16J05362)・基盤研究 (A)(26240013)、および JST 国際科学技術共同研究推進事業 (戦略的国際共同研究プログラム) 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] Avci, B., Trajcevski, G., and Scheuermann, P.: Managing evolving shapes in sensor networks, *Proc. Int. Conf. on SSDBM*, (2014).
- [2] 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).
- [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] Khan, W. Z., Xiang, Y., Aalsalem, M. and Arshad, Q.: Mobile phone sensing systems: a survey, *IEEE Communications Surveys & Tutorials*, Vol.15, No.1, pp.402–427, (2013).
- [5] 松尾 和哉, 後藤 啓介, 神崎 映光, 原 隆浩, 西尾 章治郎: 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とダイアグラムを用いた効率的な境界線検出手法, *情報処理学会論文誌*, Vol.56, No.3, pp.1107–1120, (2015).
- [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] 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, (2012).