The University of Osaka
Institutional Knowledge Archive

Title	モバイルセンサネットワークにおける境界線検出に関する研究
Author(s)	松尾, 和哉
Citation	大阪大学, 2017, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/61857
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

モバイルセンサネットワークにおける
境界線検出に関する研究

提出先　　大阪大学大学院情報科学研究科

提出年月　2017年1月

松 尾　和 哉

関連発表論文

1. 学会論文誌発表論文

1. 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 年 3 月).

2. 松尾和哉, 後藤啓介, 神崎映光, 原 隆浩, 西尾章治郎 : モバイルセンサネットワーク上の欠損データ再収集のためのエージェント制御方式, 情報処理学会論文誌, Vol. 57, No. 2, pp. 225–234 (2016 年 2 月).

2. 著書（章筆）

1. Matsuo, K., Goto, K., Kanzaki, A., Hara, T., and Nishio, S.: Data aggregation and forwarding route control for efficient data gathering in dense mobile wireless sensor networks, *Modelling and Processing for Next Generation Big-Data Technologies (Book Series: Modelling and Optimization in Science and Technologies), Springer International Publishing*, pp. 113–143 (Nov. 2014).

3. 国際会議発表論文

1. Matsuo, K., Goto, K., Kanzaki, A., Hara, T., and Nishio, S.: Data gathering considering geographical distribution of data values in dense mobile wireless sensor networks, in *Proc. Int. Conf. on Advanced Information Networking and Applications (AINA 2013)*, pp. 445–452 (Mar. 2013).

2. Matsuo, K., Goto, K., Kanzaki, A., Hara, T., and Nishio, S.: Agent redeployment and missing data regathering in mobile wireless sensor networks, in *Proc. Int.*

Computer Software and Applications Conf. (COMPSAC 2013), pp. 100–109 (July 2013).

3. Yoshimura, T., Matsuo, K., Kanzaki, A., and Nishio, S.: Integration of push-based and pull-based connectivity status sharing for efficient data forwarding toward mobile sinks in wireless sensor networks, in *Proc. Int. Conf. on Advanced Information Networking and Applications (AINA 2014)*, pp. 541–548 (May 2014).

4. Matsuo, K., Goto, K., Kanzaki, A., Hara, T., and Nishio, S.: Overhearing-based efficient boundary detection in dense mobile wireless sensor networks, in *Proc. Int. Conf. on Mobile Data Management (MDM 2014)*, pp. 225–234 (July 2014).

5. Matsuo, K., Goto, K., Kanzaki, A., Hara, T., and Nishio, S.: A study on efficient event monitoring in dense mobile wireless sensor networks, in *Proc. Int. Symposium on Reliable Distributed Systems (SRDS 2014)*, pp. 341–342 (Oct. 2014).

6. Shoji, N., Matsuo, K., Kanzaki, A., and Nishio, S.: Energy-efficient topology construction for multi-attribute data gathering in WSNs, in *Proc. Int. Conf. on Ubiquitous Information Management and Communication (IMCOM 2016)* (Jan. 2016).

7. Matsuo, K., Goto, K., Kanzaki, A., and Hara, T.: A sampling-based boundary detection method with consideration of boundary shape in dense mobile wireless sensor networks, in *Proc. Int. Conf. on Advances in Mobile Computing & Multimedia (MoMM 2016)*, pp. 307–316 (Nov. 2016).

4. 国内会議発表論文（査読有）

1. 松尾和哉, 後藤啓介, 佐々木勇和, 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データ値の地理的分布を考慮したデータ収集方式, 情報処理学会マルチメディア，分散，協調とモバイル (DICOMO2012) シンポジウム論文集, pp. 2257–2265 (2012 年 7 月).

2. 吉村 武, 松尾和哉, 神崎映光, 原 隆浩, 西尾章治郎：無線センサネットワークにおけるモバイルシンクへのデータ転送効率化のためのプッシュ型およびプル型接続状況共有手法の統合について, 情報処理学会マルチメディア，分散，協調とモバイル (DICOMO2013) シンポジウム論文集, pp. 1294–1301 (2013 年 7 月).

3. 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を用いた効率的な境界線検出手法, 情報処理学会マルチメディア通信と分散処理ワークショップ (DPSWS2013) 論文集, pp. 236–243 (2013 年 12 月).

4. 小路直紀, 松尾和哉, 神崎映光, 西尾章治郎：無線センサネットワークにおけるデータ属性ごとの観測粒度と残余電力を考慮したトポロジ構築について, 情報処理学会マルチメディア，分散，協調とモバイル (DICOMO2014) シンポジウム論文集, pp. 1539–1546 (2014 年 7 月).

5. 加藤 諒, 原 隆浩, 白川真澄, 駒井友香, 真嶋温佳, 天方大地, 大澤 純, 松尾和哉, 横山正浩, 佐崎 悠, 中村達哉, 水野聖也, 西尾章治郎：位置プライバシ問題の啓蒙のためのダミーによる位置曖昧化手法の体験システムの実装, 情報処理学会マルチメディア通信と分散処理ワークショップ (DPSWS2014), pp. 50–52 (2014 年 12 月).

6. 松尾和哉, 後藤啓介, 神崎映光, 原 隆浩：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, 情報処理学会マルチメディア，分散，協調とモバイル (DICOMO2016) シンポジウム論文集, pp. 769–776 (2016 年 7 月).

5. その他の研究会等発表論文

1. 松尾和哉, 後藤啓介, 神崎映光, 原 隆浩, 西尾章治郎：モバイルセンサネットワーク上のエージェントを用いたデータ収集における欠損データ再収集方式,

情報処理学会研究報告 (マルチメディア通信と分散処理研究会), Vol. 2012-DPS-153, No. 2, pp. 1–8 (2012 年 11 月).

2. 松尾和哉, 山本 淳, 倉沢 央, 佐藤浩史, 西尾章治郎 : 参加型センシングのためのセンサ値の変化を維持可能な精度向上手法, 電子情報通信学会総合大会, Vol. 2013, No. 583 (2013 年 3 月).

3. 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観測値の時間的変化を用いた定期的な境界線検出手法, データ工学と情報マネジメントに関するフォーラム (DEIM2015), online (2015 年 3 月).

4. 松尾和哉, 吉村 武, 神崎映光, 西尾 章治郎 : 複数シンクを活用した効率的なセンサデータ収集のためのアドホックネットワーク構築手法, 電気学会研究会資料, Vol. IS-15-61, pp. 65–70 (2015 年 11 月).

5. 小路直紀, 松尾和哉, 神崎映光, 原 隆浩, 西尾章治郎 : 無線センサネットワークにおけるセンサ端末の故障とスリープ制御を考慮した局所的なトポロジメンテナンスについて, 情報処理学会研究報告 (マルチメディア通信と分散処理研究会), Vol. 2015-DPS-165, No. 18, pp. 1–8 (2015 年 12 月).

以上

内容概要

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．これらのセンサを搭載した携帯端末の普及に伴い，一般ユーザがもつ端末が取得したセンサデータ（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利活用するユーザ参加型センシングが注目されている．また，これを実現する方法の一つとして，移動型センサ端末間の直接通信により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．

一方，収集したセンサデータを使用するアプリケーションの一つとして，観測値が一定値以上（または以下）を示す領域であるイベントの位置と形状を要求するものがある．この要求を満たすためには，各端末からセンサデータを収集し，そのセンサデータを基にイベントの境界線を推定する必要がある．トラヒックを削減しつつ，これを実現するためには，境界線付近の端末を識別し，その端末のみからセンサデータを収集することが有効である．この一連の処理を，本論文では「境界線検出」と表記する．ここで，境界線付近の端末を識別する研究はこれまでに数多く行われているが，そのほとんどが端末の位置が固定されていることを前提としているため，端末が自由に移動するモバイル環境にそのまま適用できない．

そこで本論文では，モバイルセンサネットワークにおいて，トラヒックを削減しつつ境界線を検出する手法について議論する．本研究では，端末の位置関係に基づき，近傍に位置する端末間のみで隣接関係が構築されたダイアグラムにおいて，観測値の分類が異なる端末と隣接する端末を境界線付近に位置しているとし，これを境界端末と定義する．この定義に基づき，本論文ではまず，ある時刻におけるイベントの位置と形状を要求するアプリケーションを想定し，境界線が要求された時刻における境界端末を識別する手法を提案する．次に，定期的にイベントの位置と形状を要求するアプリケーションや，制限時間内にイベントの位置と形状を要求するアプリケーションを想定し，上記の手法を基に，それぞれのアプリケーションに対応できる手法を提案する．

本論文は，5 章から構成され，その内容は次の通りである．まず第 1 章で序論を述べ，第 2 章において，ある時刻におけるイベントの位置と形状を要求するアプリケーションを想定し，ダイアグラムの逐次作成を用いて境界端末を識別する手法を提案する．この手法ではまず，クエリをフラッディングすることで，各端末にアプリケーション要求を通知すると同時に，各端末に自身の周囲に境界線が存在するかどうかを認識させる．その後，境界線の存在を認識した端末から順に，センサデータを受信しながら，そのセンサデータを基にダイアグラムを逐次作成，更新し，このダイアグラムを基に自身が境界端末であると判断した場合にのみセンサデータを送信する．また，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し，その有効性について検証した．その結果より，提案手法は，境界線の推定精度を維持しつつ，トラヒックを削減できることを確認した．さらに，端末数が増加するほど，トラヒックの削減量が増加することを確認した．

次に第 3 章において，第 2 章の手法を基に，定期的に境界線を検出できる手法を提案する．この手法では，各端末における観測値の時間的変化の監視を用いて，境界線を跨いだと判断した端末のみがセンサデータを送信し，そのセンサデータを受信した端末から順に，第 2 章の手法における境界端末の識別処理を行う．これにより，定期的にクエリをフラッディングすることなく，境界線を検出できる．また，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し，その有効性について検証した．その結果より，提案手法は，第 2 章で提案した手法と比較して，境界線の推定精度を維持しつつ，トラヒックを削減できることを確認した．さらに，端末数が増加するほど，トラヒックの削減量が増加することを確認した．

第 4 章では，制限時間内にイベントの位置と形状を要求するアプリケーションを想定し，第 2 章の手法を基に，制限時間内に収集しきれないセンサデータの数を可能な限り減らす手法を提案する．この手法では，まず境界端末を等間隔にサンプリングし，サンプリングされた端末のみがセンサデータを送信するように制御する．送信されたセンサデータを受信した他の端末は，受信したセンサデータから部分的に推定した境界線を基に，シンクへ送信するセンサデータの追加や除外を行う．また，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

を示し，その有効性について検証した．その結果より，提案手法は，第 2 章で提案した手法と比較して，境界線の推定精度を維持しつつ，制限時間内に収集しきれないセンサデータ数を削減できることを確認した．さらに，境界端末の数が極端に多い環境では，第 2 章で提案した手法より推定精度が高くなることを確認した．

第 5 章では，本論文の成果を要約した後，今後の研究課題について述べ，本論文のまとめとする．

目 次

第 1 章　序論 **1**
1.1　研究背景 1
1.2　境界線検出 3
1.3　研究内容 6
1.3.1　モバイル環境における境界端末の識別（第 2 章） 6
1.3.2　定期的な境界線検出（第 3 章） 8
1.3.3　制限時間内の境界線検出（第 4 章） 9
1.4　本論文の構成 10

第 2 章　ダイアグラムの逐次作成を用いた境界線検出手法 **13**
2.1　まえがき 13
2.2　想定環境 14
2.2.1　端末構成 14
2.2.2　アプリケーション要求 14
2.2.3　境界線推定方法 15
2.2.4　境界端末 16
2.3　関連研究 16
2.3.1　固定センサネットワークにおける境界線や等値線の検出 . . 17
2.3.2　固定センサネットワークにおける観測値の誤差を考慮した境界線の把握 19
2.3.3　モバイルセンサネットワークにおける境界線の推定 20
2.3.4　ネットワークの境界線の検出 20
2.4　提案手法 22
2.4.1　動作概要 22

2.4.2 クエリ伝播 22
2.4.3 境界端末の識別 25
2.4.4 データ収集 30
2.4.5 考察 31
2.5 性能評価 34
2.5.1 シミュレーション環境 35
2.5.2 比較手法 35
2.5.3 パラメータ 37
2.5.4 評価指標 38
2.5.5 端末数による影響 39
2.5.6 境界線の長さによる影響 42
2.6 むすび 43

第 3 章 観測値の時間的変化の監視を用いた定期的な境界線検出手法 45
3.1 まえがき 45
3.2 想定環境 46
3.3 関連研究 47
3.3.1 固定センサネットワークにおけるイベントの境界線のモニタリング 48
3.3.2 固定センサネットワークにおける等値線のモニタリング . . 49
3.3.3 固定センサネットワークにおけるイベントトに関する情報の定期的な把握 50
3.3.4 モバイルアドホックネットワークにおけるネットワークの境界線のモニタリング 51
3.4 提案手法 51
3.4.1 動作概要 51
3.4.2 境界線の存在の認識 52
3.4.3 境界端末の識別 53
3.4.4 データ収集 54
3.4.5 考察 58

3.5 性能評価 61
3.5.1 シミュレーション環境 62
3.5.2 比較手法 62
3.5.3 パラメータ 63
3.5.4 評価指標 64
3.5.5 端末数による影響 64
3.5.6 観測周期による影響 67
3.6 むすび 68

第 4 章 境界線の形状を考慮したサンプリングを用いた制限時間付き境界線検出手法 71
4.1 まえがき 71
4.2 想定環境 73
4.3 関連研究 73
4.3.1 固定センサネットワークにおける制限時間付きデータ収集 . 73
4.3.2 固定センサネットワークにおける境界線付近の端末のサンプリング 74
4.4 提案手法 74
4.4.1 動作概要 75
4.4.2 距離に基づく地理的なサンプリング 75
4.4.3 境界線の形状に応じたサンプリング 77
4.4.4 データ収集 80
4.4.5 考察 81
4.5 性能評価 82
4.5.1 シミュレーション環境 82
4.5.2 評価指標 83
4.5.3 提案手法の特性 84
4.5.4 提案手法の有効性 86
4.6 むすび 94

第 5 章　結論 **97**

5.1　本論文のまとめ 97

5.2　検討課題 99

5.2.1　リアルタイム性の高い境界線のモニタリングへの対応 100

5.2.2　実環境を想定した性能評価 100

謝辞 **103**

第1章　序論

1.1　研究背景

近年，スマートフォンやウェアラブルデバイスなど，人々が身につけて持ち歩くデバイスに様々なセンサが搭載されつつある．これらのセンサを搭載した携帯端末の普及に伴い，位置情報をはじめとした情報をセンサから定期的に取得し，これを活用するライフログ[1][2]や位置情報連動型サービス[3][4]などの普及が進んでいる．また，上記のような端末に搭載されるセンサの多様化が進み，音や温度，湿度などの環境情報を取得できる端末も普及しつつある．これらのセンサから取得した環境情報は，ライフログなどのサービスがユーザに提供できる情報を豊富にするだけでなく，多数の端末から収集することで，都市部における環境モニタリングをはじめとしたサービスにも利活用できるものと考えられる．このような背景のもと，一般ユーザが持つ端末が取得した環境情報を含むセンサデータを提供してもらい，環境モニタリングをはじめとする様々なサービスへ利活用するユーザ参加型センシングが注目されている [20, 35, 40, 48, 67, 68]．ユーザ参加型センシングでは，センサデータを提供したユーザに，収集したセンサデータから得られる有用な情報を提供するといったインセンティブを与えるなど，センサデータを提供しやすい環境を構築することで，多数のユーザからのセンサデータの収集を実現している．

[1]Lifelog, http://www.d-healthcare.co.jp/products/moveband3/index.html
[2]MOVEBAND3, http://www.d-healthcare.co.jp/products/moveband3/index.html
[3]Growth Push, https://growthpush.com/
[4]popinfo, http://www.d-healthcare.co.jp/products/moveband3/index.html

一方，近年，IEEE802.11[5] や Bluetooth[6]，ZigBee[7]，Wi-Fi Direct[8]，LTE Direct[9] などをはじめとする無線通信技術の発展と，計算機の小型化や高性能化に伴い，ルータ機能をもつ移動端末のみで一時的な無線ネットワークを形成するモバイルアドホックネットワークへの関心が高まっている [1, 25, 27, 41]．モバイルアドホックネットワークは，既存の通信インフラを必要とせずに，端末同士で自律分散的にネットワークを構築できるという特徴をもつ．この特徴を利用し，移動型センサ端末のみで構築するモバイルアドホックネットワークを用いてセンサデータを収集するモバイルセンサネットワーク [5, 31, 91] が，通信インフラを介さずにユーザ参加型センシングを実現する手段の一つとして注目されている [8, 10, 11, 16, 33, 48, 71, 79]．一般に，ユーザ参加型センシングは，LTE や WiMAX など，端末と基地局との直接通信によってインターネットに接続可能な通信インフラを介したデータ収集を想定している．しかし，このような通信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，通信インフラを介するデータ量は年々増加している [15]．一方,「現在の大阪駅周辺の気温分布が知りたい」といった，地理的に限られた範囲内のセンサデータのみが要求される場合，通信インフラを介したデータ収集をする必要はなく，上述のモバイルセンサネットワークによって，要求されるセンサデータは収集可能である．モバイルセンサネットワークでは，各端末が取得したセンサデータを，端末間の無線マルチホップ通信を用いて，センサデータを集約 · 管理するシンクへ送信する．

モバイルセンサネットワークにおけるデータ収集モデルの例を図 1.1 に示す．この図では，ノート PC（シンク）と複数のスマートフォン（端末）が無線で接続し，ネットワークを形成している．ここで，ユーザ参加型センシングにおいては，先述したアプリケーション例のように,「現在」や「大阪駅周辺」といった任意の時空間的条件を指定する要求が与えられるものと考えられる．しかし，一般ユーザの携帯する端末は，上記のような要求に関する情報をあらかじめ所持しておらず，その要求がいつ発生するのかすらも把握していないことが一般的である．そこで

[5] IEEE 802.11, http://grouper.ieee.org/groups/802/11/
[6] The Official Bluetooth Wireless Info Site, http://www.bluetooth.com/.
[7] ZigBee Alliance, http://www.zigbee.org/.
[8] Wi-Fi Peer-to-Peer, http://developer.android.com/guide/topics/connectivity/wifip2p.html
[9] LTE Direct, https://ltedirect.qualcomm.com/

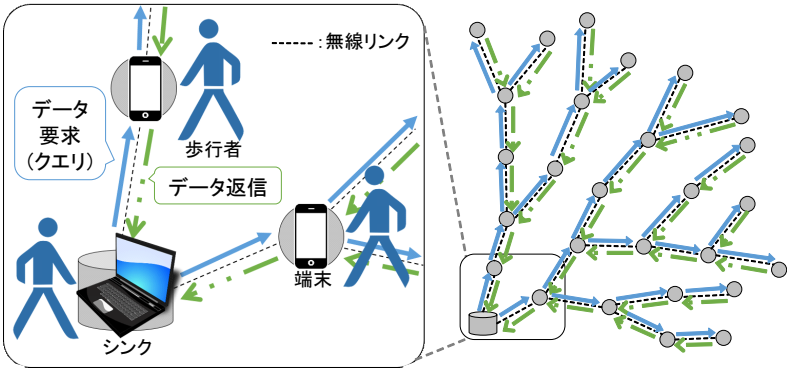


図 1.1: モバイルセンサネットワークにおけるデータ収集モデルの例

シンクは，アプリケーションが要求するセンサデータを収集するために，無線リンクを介して各端末へアプリケーション要求が含まれるデータ要求メッセージ（クエリ）を送信する．シンクと直接通信可能な端末は，クエリを直接受信し，シンクと直接通信できない端末は，シンクとの間に存在する端末の中継によってクエリを受信する．クエリを受信した各端末は，自身が搭載しているセンサを使用してセンシングを実行し，得られたセンサデータをシンクへ返信する．シンクは自身と直接通信可能な端末からデータ返信メッセージを受信する．また，シンクと直接通信できない端末からのデータ返信メッセージは，クエリと同様，間に存在する端末の中継によって，シンクへと転送される．

上記のようなセンサデータ収集では，メッセージの送受信によって，端末のバッテリや無線通信の帯域を消費する．一方，端末のバッテリや無線通信帯域は有限であるため，消費電力の抑制や無線通信帯域の有効活用は，モバイルセンサネットワークにおいて重要である．これを実現するためには，通信量（トラヒック）の削減が求められる．

1.2　境界線検出

ユーザ参加型センシングにおいて収集したセンサデータを使用する代表的なアプリケーションとして，以下に示す例のように，一定値以上（または以下）の観測値を示す領域であるイベントの位置と形状を要求するものがある．

- 日よけルート[10]の探索：各端末から収集した照度情報を基に，その値が一定値以下になる領域から日陰となっている領域を把握し，可能な限りこの領域内を通過する目的地までのルートをユーザに提示する．

- ヒートアイランド現象 [77] の調査：各端末から収集した気温情報を基に，その値が一定値以上になる領域を把握する．

- ゲリラ豪雨が気温に与える影響 [62] の調査：ゲリラ豪雨によって気温が局所的に低下した範囲を調査するために，各端末から収集した気温情報を基に，その値が一定値以下になる領域を把握する．

- 大気汚染 [29] の注意喚起：各端末から収集した PM2.5 などの微粒子濃度情報を基に，その濃度が一定値以上の領域を把握し，当該領域内に存在するユーザに対し，当該領域外への最短経路を通知する．

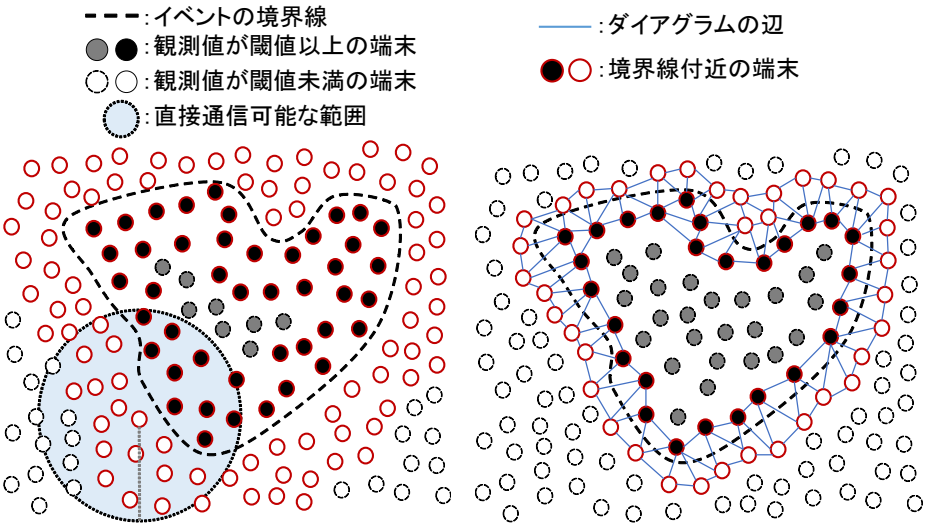
このような要求は，そのイベントの境界線を把握できれば，満たすことができる．例えば，日よけルートを検索する場合，日陰をイベントとして扱い，この境界線を把握することで，現在地点が日向の場合，最短距離でその境界線を跨ぐルートを，現在地点が日陰の場合，その境界線を可能な限り跨がないルートを，日よけルートとして提示できる．ここでモバイルセンサネットワークでは，上記の応用を含む多くの場合において，各端末が位置する地点における観測値，すなわち地理的に離散的な観測値しかわからないため，連続的な境界線を把握するためには，各端末から観測値と位置情報を含むセンサデータを収集し，そのセンサデータから境界線を推定する必要がある．

境界線の推定を実現するための単純な方法として，シンクがセンサデータを要求する地理的範囲（観測領域）内に位置するすべての端末からセンサデータを収集することが考えられる．しかし，この方法では，繁華街やテーマパークなど，端末数が多い環境において，大量のトラヒックが発生してしまう．この問題を解決するためには，境界線の推定に寄与するセンサデータのみを収集することで，トラヒックを削減する必要がある．ここで，境界線は，その境界線付近の端末のセン

[10] inShade, http://inshade.adriablue.jp/

サデータさえあれば十分に推定できることが知られている [26, 36, 38, 63, 92]．つまり，境界線から遠い端末のセンサデータを収集せず，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，トラヒックを削減しつつ，境界線を推定できる．これを実現するためには，アプリケーションがセンサデータを要求した時点において，シンクが各端末に自身が境界線付近の端末か否かを自律的に判断させ，観測領域内のすべての端末の中から当該端末を見つける（識別する）必要がある．本論文では，境界線付近の端末を識別し，それらの端末のみからセンサデータを収集する一連の処理を,「境界線検出」と表現する．また本論文では，閾値以上の観測値と閾値未満の観測値を，互いに「分類が異なる」観測値であると表現する．

ここで，境界線付近の端末を識別する研究は，端末の位置が固定されている無線センサネットワーク [6, 50, 80, 84] を対象としたものが多数存在する [26, 38, 39, 63, 75, 88]．これらの研究で提案されている手法は，境界線付近の端末をそれぞれ独自に定義し，この定義のもと，当該端末を識別している．例えば，文献 [38, 39, 88] で提案されている手法は，図 1.2(a) に示すように，直接通信可能な端末の中に観測値の分類が異なる端末が 1 台でも存在する端末を境界線付近の端末と定義している．また，文献 [26, 36] で提案されている手法は，図 1.2(b) に示すように，端末の位置関係に基づき，近傍に位置する端末間のみで隣接関係を構築するダイアグラムを予め作成し，このダイアグラム上で観測値の分類が異なる端末と隣接している端末を境界線付近の端末と定義している．ここで，図 1.2 に示すとおり，無線通信範囲に対して端末が密に存在し，直接通信可能な端末が多数存在する環境では，ダイアグラムを基に境界線付近の端末を定義することで，より境界線の近くに位置にする端末のみが，境界線付近の端末として識別されるようになる．本研究では，上記のように境界線付近の端末数の抑制に有効な，ダイアグラムを基にした後者の定義を採用し，ダイアグラム上で観測値の分類が異なる端末と隣接している端末を「境界端末」と表現する．



(a) 直接通信可能な端末を基にした境界線付近の端末

(b) ダイアグラム上で隣接する端末を基にした境界線付近の端末

図 1.2: 境界線付近の端末の定義

1.3　研究内容

本論文では，すべての端末のセンサデータを用いて推定した境界線とほぼ同じ形状の境界線を推定できる範囲で，トラヒックを削減することを目的として，モバイルセンサネットワークにおける境界線検出手法について議論する．具体的には，以下の三つの研究課題に取り組む．

1.3.1　モバイル環境における境界端末の識別（第 2 章）

文献 [26, 36] で提案されている手法は，端末が配置された時点でダイアグラムが構築されており，各端末がダイアグラム上で隣接する端末を把握している環境を想定している．そのため，各端末は，ダイアグラム上で隣接する端末と観測値を比較することで，自身が境界端末であるか否かを容易に判断できる．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，各端末の位置が時々刻々と変化し，それに伴ってダイアグラム上の隣接関係も変化する．そのため，各端末は，境界線が要求された時刻において，周辺の端末との間にどのようにダイアグラム

が構築されるかを把握できず，自身が境界端末か否か判断するために，どの端末と観測値を比較すれば良いかも把握できない．このような環境で，既存手法を用いて境界端末を識別するためには，境界線が要求された時刻において，すべての端末が自身と直接通信可能な端末と位置情報を含むメッセージを交換してダイアグラムを構築する必要があり，特に端末数が多い環境では，そのためのトラヒックが著しく増大する．

この問題を解決するために，事前にダイアグラムを構築することなく，境界線が要求された時刻における境界端末を識別できる手法を提案する．ここで，1.1 節で述べたとおり，境界線が要求された時点では，各端末はアプリケーション要求に関する情報を所持していない．そのため，境界線検出を行うためには，まずクエリを用いて各端末へアプリケーション要求に関する情報を通知する必要がある．そこで，提案手法ではまず，一部の端末のみのメッセージ転送によって，すべての端末へメッセージを伝播できる方法を用いて，アプリケーション要求を含むクエリを，すべての端末へ転送する．また提案手法では，各端末がクエリを送信（あるいは中継）する際，自身のセンサデータをクエリに付与する．これにより，クエリを受信した端末が，自身の通信範囲内に境界線が存在するかどうかを認識できる．境界線の存在を認識した端末は，自身が境界端末である可能性があることを認識し，自端末上で作成したダイアグラムを基に，自身が境界端末であるか否かを判断する．具体的には，周囲の端末が送信したセンサデータを受信しながら，センサデータに含まれる位置を頂点とするダイアグラムを逐次作成，更新し，自身が観測値の分類が異なる端末とダイアグラム上で隣接するかどうかを確認する．その結果，境界端末であると判断した場合，自身のセンサデータを送信し，そうでない場合はセンサデータの送信を中止する．また，センサデータを送信するための待ち時間を，観測値の分類が異なる端末との距離に基づいて調整することで，最終的に境界端末となる端末のみがセンサデータを送信するように制御する．これにより，センサデータを送信した端末が境界端末であるとみなすことができるため，事前にダイアグラムを構築することなく，境界端末を識別できる．最後に提案手法では，識別した境界端末のセンサデータをシンクへ送信する．

1.3.2　定期的な境界線検出（第 3 章）

イベントの位置と形状を要求するアプリケーションの中には，気温が局所的に高い領域のモニタリングなど，定期的にイベントの位置と形状を要求するものがある．このアプリケーション要求は，1.3.1 項で説明した手法による境界線検出を定期的に用いることで満足できる．しかし，この手法を定期的に用いる場合，定期的にクエリのフラッディングを行う必要があり，端末数が多い環境では，そのためのトラヒックが著しく増大する．ここで，1.3.1 項で説明した手法におけるクエリのフラッディングは，観測領域内に存在するすべての端末にアプリケーション要求を通知するために行っている．しかし，定期的に境界線検出を行う場合，各端末に 1 度だけアプリケーション要求を通知すれば，各端末は既に要求を把握している状態になるため，その後新たに端末が参加しない場合，これを定期的に行う必要はない．一方，クエリのフラッディングは，各端末に境界線の存在を認識させ，境界端末の識別処理を開始するためにも行っており，クエリをフラッディングしない場合，1.3.1 項で説明した手法による境界端末の識別が実行できなくなる．つまり，クエリをフラッディングすることなく，各端末に境界線の存在を認識させることができれば，1.3.1 項で説明した手法を定期的に用いる際のトラヒック増大を抑えつつ，効率的な境界端末の識別が可能となる．

これを実現する方法として，時間経過に伴うイベント形状の変化や端末の移動によって発生する端末の観測値の分類の変化（観測値が閾値以上から閾値未満に遷移した，またはその逆）の利用が有効であると考えられる [24, 39, 88]．これは，ある端末において観測値の分類が変化した場合，その時点において端末が境界線を跨いでおり，自身の通信範囲内に境界線が存在する可能性が高いためである．本研究では，このように観測値の分類が変化することを「観測値の時間的変化」と表現する．

この観測値の時間的変化を利用し，クエリを定期的にフラッディングすることなく，境界端末を定期的に識別できる手法を提案する．提案手法では，各端末が定期的に自身の位置における観測値を確認することで，自身の観測値の分類が変化しているか否か，すなわち自身が境界線を跨いだか否かを監視する．境界線を跨いだ端末のうち，跨いでからの経過時間が短い端末のみがセンサデータを送信し，

そのセンサデータを受信した端末から順に，境界端末の識別処理を行う．これにより，クエリを定期的にフラッディングすることなく，境界端末を識別できる．

1.3.3 制限時間内の境界線検出（第 4 章）

イベントの位置と形状を要求するアプリケーションの中には，PM2.5 や花粉などの微粒子 [29, 32] の濃度が高い領域など，毎秒十数メートル以上の速さで変化する可能性があるイベントを対象とするものもある．これらのアプリケーションは，要求した時刻の境界線が大きく変化してしまう前に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に対して，検出対象の境界線の変化の速さなどを基に，データ収集が完了するまでの制限時間を設定するものと考えられる．ここで，制限時間内にデータ収集を完了しなければならない環境では，収集可能なセンサデータ数に上限があることが知られている [2, 3, 28]．しかし，1.3.1 項で説明した手法は，制限時間が設定される要求を想定しておらず，境界端末であると判断したすべての端末からセンサデータを収集しようとしてしまう．そのため，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と，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センサ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センサデータのためのトラヒックが大量に発生し，無線通信帯域を浪費してしまう．通信帯域が有限であるモバイルセンサネットワークでは，このような無駄なトラヒックは削減するべきである．

この問題を解決するためには，制限時間内に収集できる最大センサデータ数を超えないように，データ収集を開始する前にセンサデータ数を調整すればよい．一方，収集するセンサデータを所持する境界端末の数は，検出前の境界線の長さや端末の数に依存するため，データ収集を開始する前の調整は不可能である．このような環境において無駄なトラヒックを削減するためには，端末の自律的な動作により，境界端末を識別しながら，それらの端末が送信するセンサデータの数を可能な限り削減する必要がある．さらに，アプリケーション要求の観点から，収集するセンサデータを減らしたことによって発生する，推定される境界線のずれ

を可能な限り小さくする必要もある．これらを実現するために，境界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，1.3.1 項で説明した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センサデータをシンクへ送信する．また，サンプリングされたセンサデータのみを使って推定される境界線のずれを抑えるため，送信されたセンサデータを受信した他の端末が，受信したセンサデータを用いて，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，このずれの縮小が見込める場合にのみ，シンクへ収集するセンサデータの追加や除外を行う．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，推定される境界線のずれを抑えつつ，制限時間内に収集しきれないセンサデータの送信に要するトラヒックを削減できる．最後に提案手法では，サンプリングされた境界端末のセンサデータを，制限時間内にシンクへ送信する．

1.4　本論文の構成

本論文は，5 章から構成され，本章以降の内容は次の通りである．

第 2 章では，ある時刻におけるイベントの位置と形状を要求するアプリケーションに対して，事前にダイアグラムを構築せずに，境界端末を識別する手法を提案する．この手法では，1.3.1 項で述べたとおり，クエリのフラッディング時に境界線の存在を認識した端末から順に，受信したセンサデータを基にしたダイアグラムを逐次作成し，このダイアグラムを基に自身が境界端末であると判断した場合にのみセンサデータを送信する．また，センサデータを送信するための待ち時間を調整することで，最終的に境界端末となる端末のみがセンサデータを送信するように制御する．また，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し，その有効性について検証する．

第 3 章では，イベントの位置と形状を定期的に要求するアプリケーションに対して，観測値の時間的変化を監視することで，境界端末を定期的に識別する手法を提案する．この手法では，1.3.2 項で述べたとおり，各端末における観測値の時間的変化の監視を用いて，境界線を跨いだ端末のみがセンサデータを送信し，そ

のセンサデータを受信した端末から順に，境界端末の識別処理を行う．また，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し，その有効性について検証する．

第 4 章では，制限時間内にイベントの位置と形状を要求するアプリケーションに対して，制限時間内に収集しきれないセンサデータ数を可能な限り減らすために，境界端末を地理的にサンプリングする手法を提案する．この手法では，1.3.3 項で述べたとおり，まず境界端末を等間隔にサンプリングし，サンプリングされた端末のみがセンサデータを送信するように制御する．また，送信されたセンサデータを受信した他の端末は，受信したセンサデータから部分的に推定した境界線を基に，シンクへ送信するセンサデータの追加や除外を行う．また，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し，その有効性について検証する．

最後に第 5 章では，本論文の成果を要約したのち，今後の研究課題について述べ，本論文のまとめとする．

第 2 章は文献 [56, 57, 59] で公表した結果に，第 3 章は文献 [58, 60] で公表した結果に，第 4 章は文献 [53, 54] で公表した結果に基づき論述する．

第2章　ダイアグラムの逐次作成を用いた境界線検出手法

2.1　まえがき

観測値が一定値以上を示す領域であるイベントの位置と形状を，トラヒックを削減しつつ取得するためには，1.2 節で述べたとおり，端末の自律的な動作によって境界端末を識別し，それらの端末のみからセンサデータを収集することが有効である．

境界端末の識別を実現する既存研究は，文献 [26, 36] で提案されているものをはじめとして，これまでにもいくつか行われている．しかし，これらの既存手法は，端末の位置が固定されており，各端末がダイアグラム上で隣接する端末を把握していることを前提としているため，端末の移動に伴ってダイアグラム上の隣接関係が変化する環境には対応できない．このような環境で，既存手法を用いて境界端末を識別するためには，1.3.1 項で述べたとおり，境界線が要求された時刻において，すべての端末が自身と直接通信可能な端末と位置情報を含むメッセージを交換してダイアグラムを構築する必要があり，特に端末数が多い環境では，そのためのトラヒックが著しく増大する．

本章では，この問題を解決するために，事前にダイアグラムを構築することなく，境界線が要求された時刻における境界端末を識別できる手法を提案する．1.3.1 項で述べたとおり，提案手法では，クエリのフラッディング中に境界線の存在を認識した端末から順に，受信したセンサデータを基にダイアグラムを逐次作成，更新し，自身が観測値の分類が異なる端末とダイアグラム上で隣接するか否かを確認する．その結果，境界端末であると判断した場合にのみ，観測値の分類が異なる端末に近いほど早く自身のセンサデータを送信する．これにより，事前にダイ

アグラムを構築することなく，境界端末を識別できる．

以下では，まず 2.2 節で想定環境について述べ，2.3 節で関連研究について述べる．その後，2.4 節で提案手法について述べた後，2.5 節でシミュレーション実験の結果を示す．最後に 2.6 節で本章のまとめと課題について述べる．

2.2　想定環境

本節では，本章で想定する環境について説明する．

2.2.1　端末構成

本研究では，一般ユーザの携帯するセンサを搭載した携帯端末が，互いに無線マルチホップ通信で接続している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[m] の円とする．また，この円内に位置する端末，すなわち直接通信可能な端末を，以降では「隣接端末」と表記する．各端末は GPS（Global Positioning System）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．

2.2.2　アプリケーション要求

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位置と形状を要求する．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．またシンクは，収集したセンサデータを用いて，次項で述べる方法で境界線を推定する．

2.2.3 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 [49] と呼ばれるダイアグラムを，センサデータに含まれる位置情報を基に生成して行うものとする．これは，より少ないセンサデータ数で，境界線を推定するためである．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，センサデータに含まれる位置を頂点としたドロネー三角形を生成し，これに基いて，境界線を推定する．具体的には，センサデータに含まれる位置を頂点とする三角形のうち，その外接円内に他のセンサデータに含まれる位置が存在しないものを選出する．これらの三角形の辺（ドロネー辺）の中で，観測値の分類が互いに異なる地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 2.1 は，上記の方法によって推定された境界線の例を示しており，黒丸は観測値が閾値以上の位置，白丸は観測値が閾値未満の位置を示す．このように，境界線の推定にドロネー図を用いることにより，推定に必要なセンサデータが，観測値の分類が異なる端末と隣接する端末のもののみに限定される．そのため，すべての端末からセンサデータを収集する場合と比較して，収集するセンサデータ数を大幅に削減できる．

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，実際の境界線と表現する．上記の方法によって，どれほど実際の境界線に近い境界線を推定できるかは，どれほど実際の境界線に近い位置の値を観測できるかに依存する．具体的には，観測値と位置情報に誤差がない場合，環境情報の値を観測できる位置が実際の境界線に近づくほど，互いに観測値の分類が異なる 2 地点間を結ぶドロネー辺の中点が，実際の境界線上に位置する確率が高くなる．つまり，環境情報の値を観測できる位置が多い，すなわち端末密度が高い環境であるほど，実際の境界線に近い境界線を推定できる．

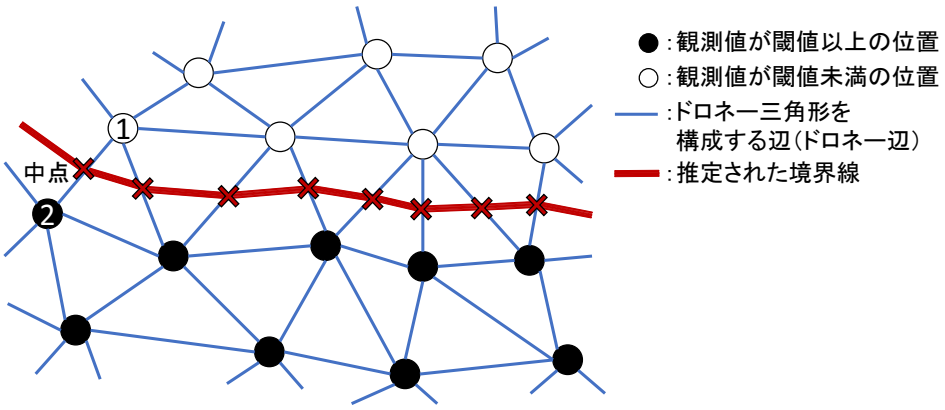


図 2.1: 境界線推定方法

2.2.4　境界端末

本研究では，すべての端末の位置を頂点とするドロネー図上で，観測値の分類が異なる端末と隣接する端末を「境界端末」と定義する．すなわち，前項で述べた推定方法で用いられるセンサデータを所持する端末が境界端末となる．例えば図 2.1 において，位置 1 と 2 にいた端末は，互いに観測値の分類が異なり，かつドロネー図上で隣接するため，境界端末となる．提案手法は，この定義に基づく境界端末を識別し，それらの端末のみからセンサデータを収集する手法である．つまり提案手法は，境界線の推定に用いられるセンサデータの条件に依存する手法である．

2.3　関連研究

本節では，端末の位置が固定されている固定センサネットワークにおけるイベントの境界線や等値線の検出，観測値の誤差を考慮した境界線の把握，およびモバイルセンサネットワークにおける境界線の推定に関する既存研究について述べる．また，ネットワークの境界の検出に関する既存研究についても述べる．

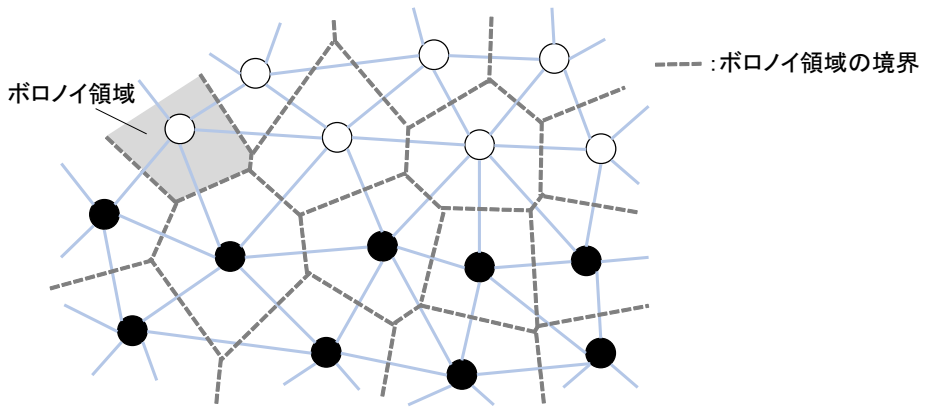


図 2.2: ボロノイ図

2.3.1 固定センサネットワークにおける境界線や等値線の検出

観測値の分類が異なる端末のセンサデータを用いた境界線の検出

文献 [26, 36] では，各端末が自身とドロネー図上で隣接する端末を予め把握している環境において，各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．ボロノイ図とは，2.2.3 項で述べたドロネー図と双対の関係にあるダイアグラムであり，図 2.2 に示すとおり，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る．文献 [26] で提案されている手法では，ボロノイ図上で観測値の分類が異なる端末と隣接する端末のセンサデータのみを収集し，それらの端末の位置を母点によって形成されるボロノイ領域の境界を観測値の境界線として推定している．また文献 [36] では，境界線の位置が時間的に変化する環境にも対応できる手法を提案している．これらの研究は，ダイアグラムを用いてイベントの境界線を検出している点が本研究と類似している．

文献 [63] では，観測領域を 4 分木を用いて分割および階層化し，境界線付近のセンサデータのみを細かい粒度で収集する手法を提案している．この手法では，分割された各領域をクラスタとして扱い，各クラスタの代表として選出したクラスタヘッドを節点とする集約木が予め構築されている環境において，この集約木における葉ノードに当たるクラスタヘッドから順にセンサデータを収集する．このとき，自身のクラスタ内に境界線が存在しない，すなわち観測値の分類が互いに

異なるセンサデータを受信しない場合，集約木上の子から受信したセンサデータを一つのセンサデータに集約し，集約木上の親へ送信する．これにより，境界線付近のセンサデータのみを細かい粒度で取得できる．この研究は，境界線付近のセンサデータを多く収集するという点が本研究と類似している．

文献 [23, 73, 74] で提案されている手法では，予め構築されたセンサデータ収集用の木構造ネットワークにおいて，境界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，トラヒックを抑えている．これらの研究は，境界線付近に位置する端末のみからセンサデータを収集する点が本研究と類似する．

上記の研究では，境界線の検出をするため，専用の固定端末を観測領域内に配置している環境を想定している．これらの既存研究とは異なり，本研究では，センサ端末として一般ユーザの所持する携帯端末を想定しているため，各端末はアプリケーションの要求を事前に知らない．また，各端末は自由に移動するため，自身の周辺に位置する端末についても把握していない．そのため，各端末が要求および自身の隣接端末を把握していることを前提とした既存研究のアプローチを，そのまま適用できない．

凸包を用いた境界線の検出

文献 [22, 51, 70] では，観測値が閾値以上の端末を包含する凸包を用いて境界線を検出する手法を提案している．文献 [51, 70] で提案されている手法では，各端末が受信した位置情報と自身の位置情報を基に，部分的に凸包を作成し，その頂点に当たる位置情報のみを送信するという処理を繰り返しながら，凸包の頂点に位置する端末を識別する．また文献 [22] では，位置情報が不要な手法を提案している．この手法では，凸包が存在し得る範囲を計算し，その範囲内でのみメッセージをフラッディングする．このときのメッセージの送信ホップ数と端末間の隣接関係を利用することで，位置情報なしで凸包の頂点に位置する端末を探索する．

これらの研究では，観測値が閾値以上の端末を包含する凸包を境界線として推定しており，その境界線上に位置する端末のみを識別している点が本研究と類似している．しかし，これらの既存研究で採用している境界線付近の端末の定義は，

本研究で採用しているものと異なり，観測値の分類が異なる端末のセンサデータを用いないものになっているため，これらの既存研究のアプローチはそのまま適用できない．

等値線の検出

観測値の等値線の位置と形状を要求するアプリケーションを想定した研究も存在する [52]．この研究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のみが，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と，自身のセンサデータをシンクへ返信する．シンクは受信したセンサデータと観測値の傾きから，等値線を推定する．

この研究は，等値線付近に位置する端末を識別している点が本研究と類似する．一方，等値線付近の端末の定義は，本研究で採用しているものとは異なり，観測値の分類が異なるセンサデータを用いないものであるため，この既存研究のアプローチはそのまま適用できない．

2.3.2 固定センサネットワークにおける観測値の誤差を考慮した境界線の把握

観測値に含まれる誤差により，境界線による観測値の明確な分類が困難な環境において，境界線の位置を把握する研究も存在する [13, 37, 83]．文献 [13] で提案されている手法では，各端末が自身から k ホップの範囲内にいる端末からセンサデータを収集し，それらのセンサデータを用いて境界線の位置を推定する．推定した境界線が自身の近くにある場合，その端末は境界線付近に位置すると判断する．文献 [37] で提案されている手法では，各端末が自身の隣接端末の観測値を収集し，その平均値を算出し，その値を基に自身が境界線付近に位置するか否かを判断する．またこれらの研究は，すべての端末から，自身が境界線付近に位置するか否かに関する情報を収集し，その分布を確認することで，境界線のおおよその位置を把握する．文献 [83] で提案されている手法では，各端末がボロノイ図上で隣接する端末の観測値を用いて，自身の観測値を修正する．その後，すべての

端末からセンサデータを収集し，サポートベクターマシンを用いて，観測値が閾値以上の地点と未満の地点を分類する線を引く．

これらの研究は，イベントの位置と形状を要求するアプリケーションを対象としている点が本研究と類似しているが，観測領域内のすべての端末から情報を収集する点と，観測値の誤差を考慮している点が本研究と異なる．なお，観測値や位置情報に含まれる誤差により，境界線による観測値の明確な分類が困難な環境で，提案手法がどのように動作するかについては，2.4.5 項において議論する．

2.3.3　モバイルセンサネットワークにおける境界線の推定

移動型端末を用いて境界線を推定する研究もこれまでにいくつか行われている [17, 24, 72, 76]．文献 [17, 72, 76] では，移動が制御できる端末を用いて境界線を推定する手法を提案している．これらの手法では，端末に境界線上を移動させ，その軌跡を取得することで，境界線を推定している．これらの研究は，移動型端末を用いて境界線を推定している点が本研究と類似しているが，端末の移動を制御できる点が異なる．

一方，文献 [24] では，領域内を自由に移動する端末を用いて境界線を推定する手法を提案している．この手法では，境界線を跨いだ端末とその付近に位置する端末のセンサデータを収集することで，部分的に境界線を推定する．この処理を繰り返すことで，境界線の全体の形を推定する．この研究は，移動が制御できない端末を用いて静的な境界線の検出を行っている点が本研究と類似しているが，長い時間をかけて少しづつ境界線を推定する環境を想定している点が異なる．

2.3.4　ネットワークの境界線の検出

イベントの境界線に限らず，ネットワークを形成する端末が存在する範囲であるネットワークの境界を検出する研究もこれまでにいくつか行われている [18, 44, 45, 81, 90]．文献 [18, 81] では，固定センサネットワークにおいて，位置情報なしでネットワークの境界線を検出する手法を提案している．文献 [18] で提案されている手法では，端末の隣接関係を辺としたグラフから，ネットワーク全体のスケ

ルトングラフを構築し，これを用いてネットワークの境界線に位置する端末を識別する．文献 [81] で提案されている手法では，ある端末からフラッディングしたメッセージのホップ数を用いて，ネットワークの中心付近に位置する端末群を探索する．その端末群からさらにフラッディングしたメッセージのホップ数が最大となる端末群を，ネットワークの境界線に位置する端末として識別する．文献 [90] で提案されている手法では，3 次元空間上に構築された固定センサネットワークにおいて，各端末を母点とするボロノイ図を用いてネットワークの境界面を作成し，その外側に位置する端末をネットワークの境界に位置する端末として識別する．これらの研究は，すべての端末が自身の隣接端末を把握している環境を想定しており，端末の移動により，その隣接関係が変化する環境を想定していない点が本研究と異なる．

文献 [44, 45] では，モバイルアドホックネットワークにおいて，すべての端末を包含する凸包を用いてネットワークの境界線を検出する手法を提案している．この研究では，クエリ発行端末から最も遠い端末から順に，部分的に凸包を計算しながら，その頂点に当たる位置情報のみを収集する手法と，隣接する端末間の角度を用いて，凸包の頂点に当たる端末にだけメッセージを送信しながら，その端末の位置情報のみを収集する手法を提案している．この研究は，端末の移動を考慮しており，各端末が自身の隣接端末を把握していない環境を想定している点が本研究と類似している．

上記の研究では，ネットワークの境界を要求するアプリケーションを対象としている．ここで，観測値が閾値以上の端末でのみ形成されるネットワークの境界線に位置する端末を，イベントの境界線付近の端末と定義した場合，上記の既存研究のアプローチを用いて，境界線付近の端末を識別できる．つまり，これらの研究は，ネットワークの境界に位置する端末を識別する点において本研究と類似している．しかし，これらの既存研究で採用している境界付近の端末の定義は，本研究で採用しているものと異なり，観測値の分類が異なる端末のセンサデータを用いないものになっているため，これらの既存研究のアプローチはそのまま適用できない．

2.4 提案手法

本節では，まず提案手法の概要について述べ，その後，提案手法の詳細な手順について説明する．

2.4.1 動作概要

2.2.1 項で述べたように，本研究で想定する環境では，各端末はアプリケーションの要求を把握していない．これらの端末を用いて境界線を検出するためには，観測領域内にいる全端末にその要求を通知しなければならない．そのため，まず提案手法では，図 2.3 に示すように，アプリケーションの要求をクエリとして，2.4.2 項で述べる方法で観測領域全体に伝播する．

提案手法では，クエリを転送する端末がクエリに自身の観測値を載せて転送しながら，クエリを伝播する．これにより，各端末は通信範囲内においてクエリを転送した端末の観測値を把握でき，分類が異なる観測値を含むクエリを受信した端末は，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．例えば図 2.3 において，端末 1 がクエリに自身の観測値を載せて転送することで，このクエリを受信し，端末 1 とは分類が異なる値を観測した端末 2 は，自身と端末 1 との間に境界線が存在することがわかる．このように，自身の通信範囲内に境界線が存在することを認識した端末から順に，2.4.3 項で述べる方法に従って，自身が境界端末か否かを判断し，観測領域内の端末から境界端末を識別する．

最後に，境界端末が送信したセンサデータを，2.4.4 項で述べる方法で収集する．

2.4.2 クエリ伝播

クエリ伝播に要するトラヒックを削減するために，端末およびシンクは，文献 [78] 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，一部の端末のみのクエリ転送によって，すべての端末へクエリを伝播する．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，転送端末の識別

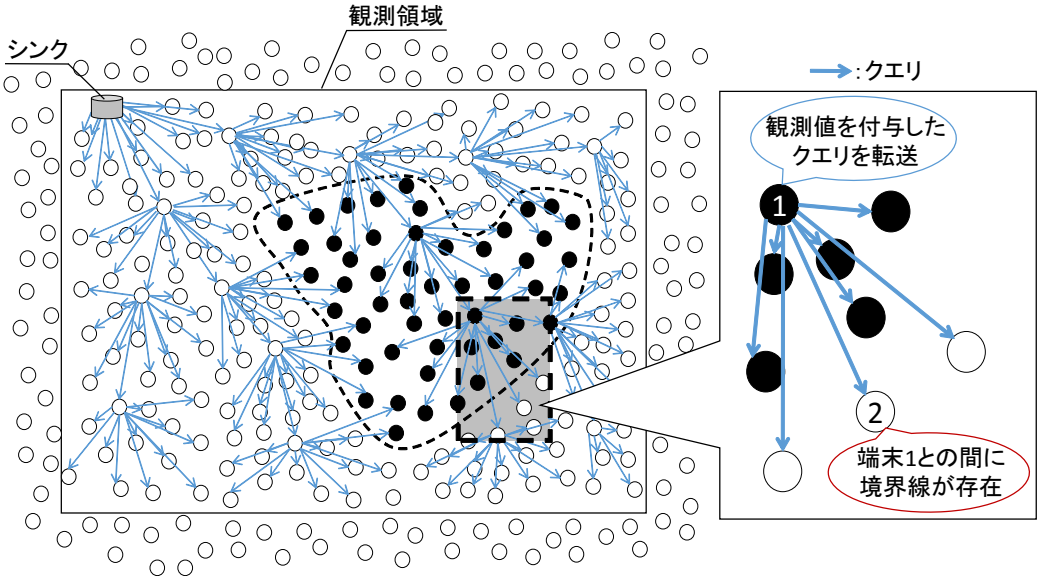


図 2.3: クエリの伝播と境界線の存在の認識

子，位置，取得した観測値を含み，後述する境界端末の識別処理の終了条件として，識別終了時刻を含む．さらに，後述するデータ収集時に用いる情報として，シンクからのホップ数と，シンクからのホップ数の最大値を含む．ここで，シンクからのホップ数の最大値は，シンクから最も遠い観測領域端の位置，および無線通信範囲を用いて，シンクが設定する値である．

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．このときクエリ内に含まれるシンクからのホップ数は 0 に設定する．クエリを受信した各端末は，自身が観測領域内に位置している場合，センシングおよび自身の位置情報の取得を行い，センサデータを生成する．クエリを受信した各端末は，まずクエリ転送のための待ち時間 W_q を，次式に従って設定する．

$$W_q = \left(1 - \frac{l}{r}\right) \cdot W_q^{max}. \tag{2.1}$$

ここで，l はクエリの転送元端末との距離を示す．W_q^{max} は待ち時間 W_q の最大値であり，全端末共通で予め設定される値である．式 (2.1) より，待ち時間 W_q 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．待ち時間を設定した端

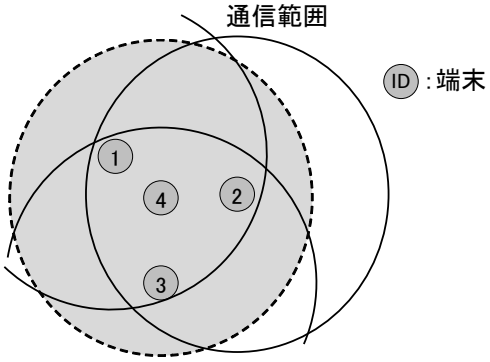


図 2.4: クエリの転送を中止する端末の例

末は，待ち時間内に他の端末が転送したクエリを受信し，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，クエリ内の転送端末に関する情報を自身のものに更新し，シンクからのホップ数を 1 だけ増加させた後，クエリを転送する．図 2.4 は，クエリの転送を中止する端末の例を示している．この図では，端末 4 の通信範囲全体が観測領域内に存在するため，クエリが指定する観測領域とその通信範囲が重なる領域は，端末 4 の通信範囲全体となる．端末 4 がクエリを転送する前に，端末 $\{1, 2, 3\}$ がクエリを転送した場合，端末 4 の通信範囲全体が端末 $\{1, 2, 3\}$ の通信範囲によって被覆されている．よって，端末 4 はクエリの転送を中止する．また，クエリを既に受信している端末は，新たに受信したクエリを破棄し，その転送は行わない．これにより，一部の端末のみのクエリ転送によって，すべての端末へクエリを伝播できる．

各端末は，自身にクエリを転送した端末の識別子を記録し，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内部ノードとする木構造ネットワークである転送木を構築する．ここで各端末は，自身にクエリを転送した端末を，転送木上の親として認識する．また，転送したクエリに設定したシンクからのホップ数を，転送木上における自身からシンクまでのホップ数として記録する．データ収集時には，この転送木を用い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．これらの観測値の分類が異なる場合，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端末である可能性があると判断し，次項で述べる境界端末の識別処理を開始する．このように提案手法では，本章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，自身の通信範囲内に境界線が存在する端末のみで境界端末の識別処理を行う．これにより，境界線から遠い領域でのセンサデータの交換が不要となり，トラヒックの増大を抑制できる．

2.4.3 境界端末の識別

提案手法では，各端末が受信したセンサデータを基にダイアグラムを逐次作成しながら，自身が境界端末か否かを判断する．ここで，2.2.3 項で述べたドロネー三角形を作成するためには，センサデータの数が三つ以上必要であるため，ある端末が自身のセンサデータ以外に一つしかセンサデータを受信していない場合，自身が境界端末か否かを判断できない．この問題を解決するために，提案手法では，2.3.1 項で述べたボロノイ図を用いる．ボロノイ図とドロネー図は双対の関係にあるため，ドロネー図上で隣接する端末とは，ボロノイ図上でも隣接する．またこのボロノイ図は，その定義上，母点となる位置の数が一つしかない場合においても作成できる．

境界端末の識別処理における端末 n の動作を Algorithm1 に示す．なお，説明の都合上，アルゴリズム内では表 2.1 に示す記号を用いている．表 2.1 における $DN(n)$ および $CDN(n)$ の例を，図 2.5 に示す．この図において，端末 1 とボロノイ図上で隣接し，かつ端末 1 と観測値の分類が異なる端末群 $DN(1)$ は，端末 $\{2, 3, 4\}$ であり，その中で最も端末 1 に近い $CDN(1)$ は，端末 2 である．

観測値の分類が異なるクエリ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，およびクエリ受信時に取得した自身の位置情報を基にボロノイ図 $V(n)$ を作成する（7，20 行目）．

表 2.1: 2.4.3 項で用いる記号

記号	意味
Q_{dif}	分類の異なる観測値が含まれたクエリ
SD_{new}	新たに受信したセンサデータ
$SD_{received}(n)$	端末 n が受信したセンサデータ集合
$V(n)$	端末 n が構築したボロノイ図
$DN(n)$	ボロノイ図上で隣接する観測値の分類が異なる端末群
$CDN(n)$	$DN(n)$ の中で最も近い端末
$CDN_{prev}(n)$	SD_{new} を受信する前の $CDN(n)$
$T_{boundary}$	タイマ
t_{detect}	識別終了時刻

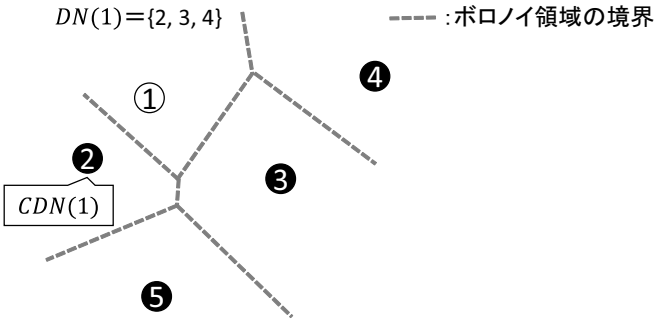


図 2.5: $DN(n)$ および $CDN(n)$ の例

2. ボロノイ図上で隣接し，かつ観測値の分類が異なる端末が存在し（$DN(n) \neq \emptyset$），さらに $DN(n)$ の中で最も自身に近い端末 $CDN(n)$ が，手順 (1) を実行する前と異なる（$CDN(n) \neq CDN_{prev}(n)$）場合（24 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 $T_{boundary}$ を設定する（25 行目）．

$$T_{boundary} = \frac{d}{r} \cdot T_{boundary}^{max}. \tag{2.2}$$

上式の d は $CDN(n)$ との距離である．$T_{boundary}^{max}$ はタイマ $T_{boundary}$ の最大値であり，全端末共通で予め設定される値である．式 (2.2) 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，同様にタイマ $T_{boundary}$ 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，クエリ受信時に取

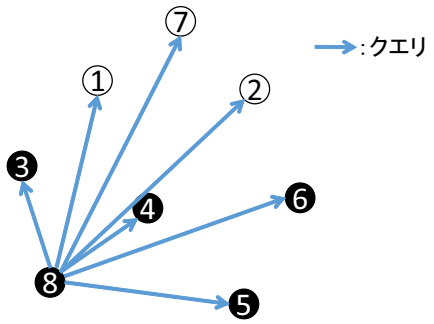


図 2.6: クエリ転送の例

得した自身のセンサデータを，2.4.2 項で構築した転送木上の親へ送信し，境界端末の識別処理を終了する（14，15 行目）．

4. $DN(n)$ が存在しない場合，センサデータの送信は行わない．また，新たに受信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（29 行目）．

5. シンクが設定した識別終了時刻 t_{detect} までに新たにセンサデータを受信した場合は，手順 (1) に戻る（17 行目）．

提案手法における各端末の動作として，図 2.6 のように，端末 8 が送信したクエリを，端末 $\{1, \ldots, 7\}$ が受信した場合を想定した手順を，図 2.7 を用いて示す．図 2.7 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し，吹出しは，各端末がそれぞれの時点で作成しているボロノイ図を示す．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受信可能であるものとする．

1. 端末 $\{1, \ldots, 7\}$ は，まず端末 8 を転送木上の親として記録する．また，この時点では，各端末は自身と端末 8 の位置と観測値しか認識していないため，それぞれ自身と端末 8 のみを母点としたボロノイ図を作成する．例えば図 2.7(a) において，端末 $\{1, 2\}$ がそれぞれ自身と端末 8 のセンサデータを用いてボロノイ図を作成している．図からわかるとおり，端末 $\{1, 2\}$ はボロノイ図上で端末 8 と隣接し，互いの観測値の分類が異なるため，端末 8 との距離に応じ

Algorithm 1 提案手法における境界端末の識別

```
SD_received(n) ← ∅
DN(n) ← ∅
CDN(n) ← null
CDN_prev(n) ← null
T ← ∞
if Received query is Q_dif then
    Create V(n) using its sensor data and sensor data contained in Q_dif.
    DN(n) ← Location of the node which transmitted Q_dif
    CDN(n) ← Location of the node which transmitted Q_dif
    Set T_boundary.
end if
while t_detect does not come do
    if Timer T_boundary expired then
        Send its sensor data to its parent on the forwarding tree.
        break
    end if
    if Received SD_new then
        SD_new is added to SD_received(n).
        CDN_prev(n) ← CDN(n)
        Update V(n) using its sensor data and SD_received(n).
        DN(n) ← Locations of nodes with a different sensor reading
                    which are adjacent in V(n)
        if DN(n) ≠ ∅ then
            CDN(n) ← Location of the closest node in DN(n)
            if CDN(n) ≠ CDN_prev(n) then
                Set T_boundary.
            end if
        else
            CDN(n) ← null
            T_boundary ← ∞ // Cancelling the timer
        end if
    end if
end while
```

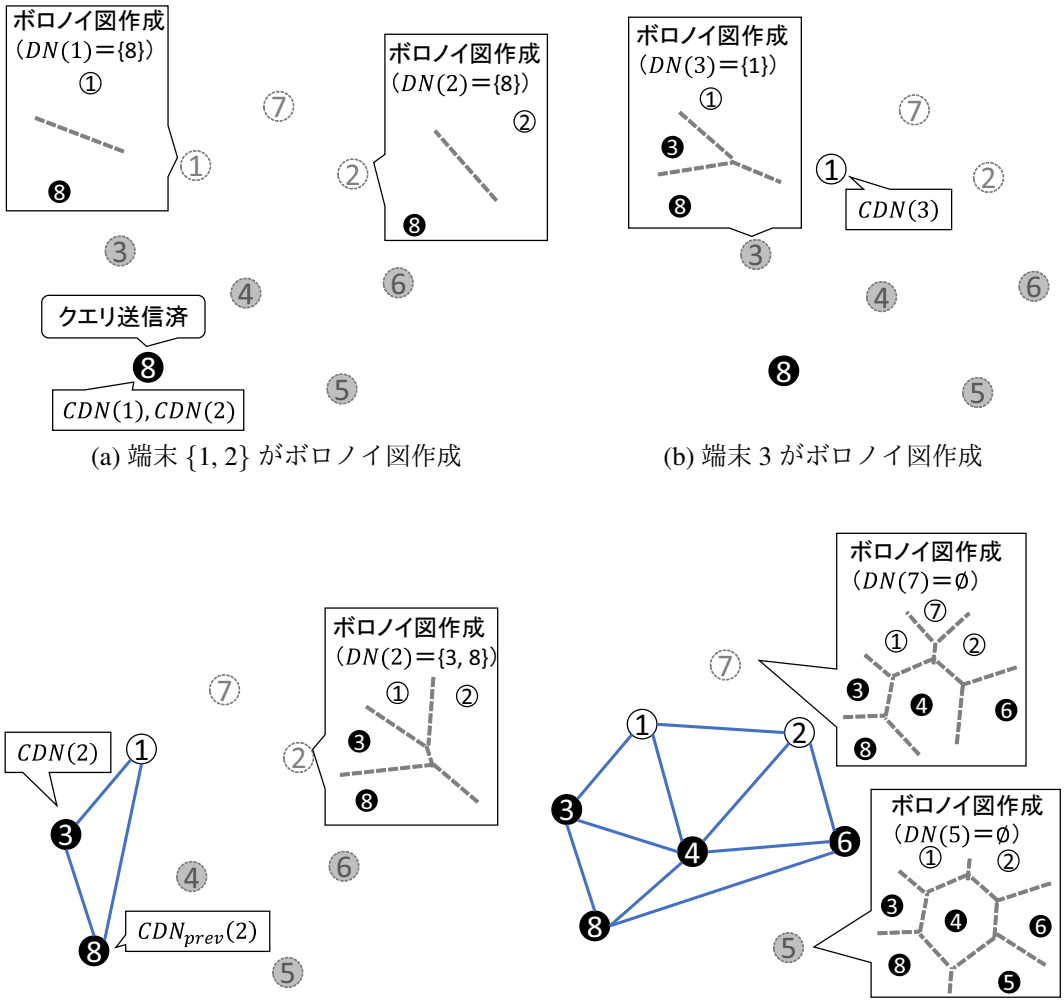
たタイマを設定する．ここで，端末 2 より端末 1 のほうが端末 8 に近いため，端末 1 のタイマが先に切れ，転送木上の親である端末 8 へセンサデータを送信する．

2. このセンサデータを受信した端末 $\{2, \ldots, 8\}$ は，端末 1 の存在および観測値を認識し，この時点で受信している端末 $\{1, 8\}$ のセンサデータを基に，端末 $\{1, 8\}$ と自身のみを母点とするボロノイ図を作成する．例えば端末 3 は，図 2.7(b) に示すボロノイ図を作成しており，この図において自身と観測値の分類が異なる端末 1 と隣接しているため，端末 1 との距離に応じたタイマを設定する．端末 $\{4, 5, 6, 8\}$ 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 8 へセンサデータを送信する．

3. 端末 3 のセンサデータを受信した端末のうち，センサデータを送信していない端末 $\{2, 4, \ldots, 8\}$ は，自身と端末 $\{1, 3, 8\}$ のみを母点とするボロノイ図を作成する．例えば端末 2 は，図 2.7(c) に示すボロノイ図を作成する．ここで，端末 2 はボロノイ図上で自身と隣接し，かつ観測値の分類が異なる端末 $DN(2) = \{3, 8\}$ の中で，自身に最も近い端末 $CDN(2)$ が端末 8 から端末 3 に変わるため，端末 3 との距離に応じたタイマを再設定する．

4. 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 8 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受信した端末 5 は，図 2.7(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 がボロノイ図上で隣接する端末は $\{4, 6, 8\}$ であり，観測値の分類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．端末 7 も同様に，観測値の分類が異なる端末が存在しない（$DN(7) = \emptyset$）ため，センサデータの送信を中止する．

上記の処理により，クエリを転送した端末 8 を除き，ボロノイ図上で観測値の分類が異なる端末と隣接する端末 $\{1, 2, 3, 4, 6\}$ のみが，センサデータを送信できる．



(a) 端末 $\{1, 2\}$ がボロノイ図作成

(b) 端末 3 がボロノイ図作成

(c) 端末 2 がボロノイ図作成

(d) 端末 $\{5, 7\}$ がボロノイ図作成

図 2.7: 提案手法における境界端末識別の例

2.4.4　データ収集

識別終了時刻になった時点で，2.4.2 項でクエリを転送した端末，すなわち転送木の内部ノードに該当する端末は，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，一つのメッセージに添付して，転送木上の親へ送信する．メッセージを受信した端末（親）は，メッセージを送信した子へ受領確認（ACK）を送信する．ACK を受信しなかった端末（子）は，親の ACK

が受信できるまで一定回数メッセージを再送する．自身がメッセージを送信した後に，自身の子からメッセージを受信した場合は，そのメッセージをそのまま自身の親へ送信する．収集用の待ち時間 W_g は，次式に従って設定する．

$$W_g = \{(h_{max} - h) + \mathit{rand}\} \cdot W_g^{max}. \tag{2.3}$$

ここで，h は自身に設定したシンクからのホップ数，h_{max} はクエリに付与されていたシンクからのホップ数の最大値を示す．これらの情報は，2.4.2 項で述べたクエリ伝播時に把握済みである．また，*rand* はパケット衝突を回避するために与える乱数である．W_g^{max} は，待ち時間 W_g の最大値であり，全端末共通で予め設定される値である．

式 (2.3) より，待ち時間 W_g は，シンクからのホップ数が多いほど短く設定される．この待ち時間に従ってメッセージを送信することで，シンクから遠い端末から順にメッセージが送信され，最終的に，2.4.3 項で述べた識別処理において送信された全センサデータが，シンクに収集される．

2.4.5 考察

提案手法では，受信したセンサデータを基に各端末がボロノイ図を逐次作成し，自身が境界端末か否かを判断する．そして，境界端末と判断した端末のみが，自身と観測値の分類が異なる端末に近いほど短くなるタイマを設定してセンサデータを送信することで，境界端末のみがセンサデータを送信するように送信順序を制御する．これにより，事前にドロネー図を構築することなく，各端末の自律的な動作によって境界端末を識別できる．本項では，境界端末が極端に多い環境，境界端末が存在しない環境，観測値や位置情報に誤差が含まれている環境において，この提案手法がどのように動作するのかについて議論する．また，提案手法がデータ収集を完了するまでに要する時間について，1.2 節で述べたような，すべての端末から個別にセンサデータを収集する手法と比較しつつ，議論する．

提案手法の挙動について

境界端末数が極端に多い環境：提案手法では，自身が境界端末であると判断したすべての端末からセンサデータを収集するため，境界端末数が極端に多くなる環境では，境界端末のみからセンサデータを収集するとはいえ，大量のトラヒックが発生してしまう．このような，収集データ数を制御できないことによって発生する問題については，第 4 章で取り組む．

境界端末が存在しない環境：検出対象のイベントが観測領域内に存在しない環境や，検出対象のイベント内に端末が存在しない環境では，そもそも境界端末が存在しないため，提案手法は動作しない．このような環境では，提案手法に限らず，2.3 節で述べたイベントの境界線を検出する既存手法を用いたとしても，境界線は検出できない．一方，イベント内に端末が存在しない場合，当該イベント周辺において，どの端末の通信範囲にも被覆されない範囲（カバレッジホール）が発生している場合があるため，2.3.4 項で述べたネットワークの境界線を検出する手法を用いることで，当該イベントのおおよその位置と大きさを把握できる可能性がある．

観測値や位置情報に誤差が含まれている環境：また，2.3.2 項で述べたとおり，提案手法は観測値や位置情報に誤差が含まれている環境を考慮していない．観測値に誤差が含まれる環境では，本来閾値以上の値を観測するはずであった端末が，閾値未満の値を観測する状況が多発する [13, 37, 46] 場合がある．この場合，図 2.8 に示すように，互いに分類が異なる観測値が境界線の周辺で混在してしまい（混在領域），各地点の観測値の明確な分類が難しくなる．このような状況は，各端末が取得した位置情報が誤差 [85, 86, 87] によってずれた場合にも発生する．具体的には，閾値以上の値を観測した端末が，測地装置の測定誤差によって，観測値が閾値未満の値を示す領域内の位置を取得した場合，観測値に誤差が生じる場合と同様に，混在領域が発生する．

しかし，上記のように混在領域が発生する環境においても，提案手法ではボロノイ図上で観測値の分類が異なる端末と隣接する端末しかセンサデータを送信しないため，図 2.9 に示すように，混在領域から離れた位置にいる端末はセンサデータを送信しない．すなわち，提案手法は，観測値や位置情報に誤差が含まれてい

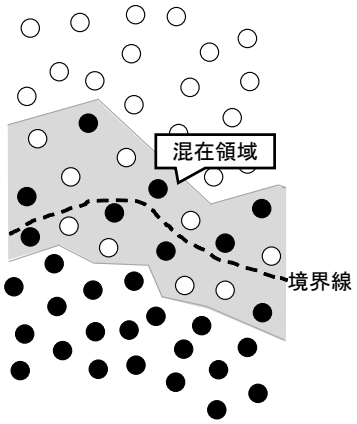


図 2.8: 各地点の観測値の明確な分類が困難な環境の例

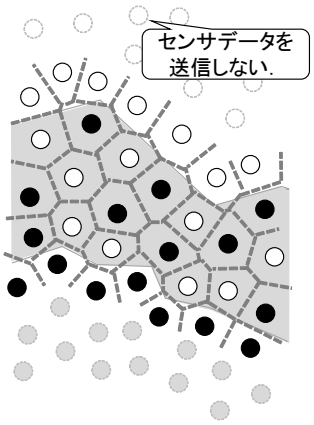


図 2.9: 図 2.8 の環境における提案手法による境界端末の識別

る環境においても，すべての端末が自身の隣接端末が事前に位置情報を含むメッセージを交換する手法より，少ないトラヒックで境界端末を識別できる．

データ収集が完了するまでに要する時間について

提案手法において，境界線検出に要する時間は，2.4.2 項で述べた識別終了時刻と，2.4.4 項で述べたデータ収集時の待ち時間の設定に依存する．また，式 (2.3) における h_{max} は，実際の最大ホップ数（h_{max}^{ideal}）がわからないシンクが設定する値であるため，実際の値より大きく設定される．さら，パケット衝突を避けつつデータ収集を行うためには，式 (2.3) における待ち時間の最大値 W_g^{max} を大きく設定する必要がある．一方，1.2 節で述べたような，すべての端末から個別にセンサデータを収集する手法では，上記のような設定が必要ないため，環境によっては，この手法のほうが，トラヒックは大きくなるものの，データ収集が完了するまでの時間が短くなる可能性がある．

具体的には，すべての端末から個別にセンサデータを収集する手法において，データ収集が完了するまでの時間は，シンクから最も遠い端末からシンクまでの実際のホップ数 h_{max}^{ideal}（$< h_{max}$），および 1 ホップあたりの転送遅延のみに依存する．パケット衝突が発生しにくい環境では，下位層におけるデータフレームの再送が発生しにくくなるため，1 ホップあたりの転送遅延が小さくなり，結果として

データ収集に要する時間が短くなる．これは，収集するセンサデータ数，すなわち観測領域内の端末数が少ないほど起こりやすいものと考えられる．一方，端末数が多い場合，データ収集時のメッセージ数が膨大になり，トラヒックが著しく増大してしまう．そのため，通信帯域が圧迫されてしまい，下位層におけるデータフレームの再送の頻発によって，1 ホップあたりの転送遅延が大きくなり，結果として，提案手法よりデータ収集が完了するまでに要する時間が長くなる場合があるものと考えられる．また，データフレームの再送回数が一定値に達してパケットロスが発生し，結果として境界線の推定に寄与するセンサデータがロスする場合もあるものと考えられる．

一方，提案手法は，パケット衝突を避けつつ，境界線の推定に寄与するセンサデータのみを収集できるため，端末数が多い環境では，すべての端末から個別にセンサデータを収集する手法より，効率的にセンサデータを収集できるものと考えられる．また，提案手法におけるデータ収集を非常に短い時間で実現しようとしたときに発生する問題については，第 4 章で取り組む．

2.5　性能評価

本節では，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，ある時刻におけるイベントの境界線が求められる環境における提案手法の性能を評価するために，テーマパークなど歩行者の通行を阻害するものが極めて少ない環境において，テーマパーク事業者が，ミストシャワー等の熱中症対策機器を設置するために，ある時刻において気温が局所的に高い領域の境界線を検出する状況を想定した．本実験では，ネットワークシミュレータ Scenargie 2.0[1] を用いた．

[1]Scenargie2.0 Base Simulator revision 18021, Space-Time Engineering, https://www.spacetime-eng.com/

2.5.1 シミュレーション環境

観測領域のサイズは，会場の一部であり，注意喚起を行う範囲を包含可能な大きさとして，1,000[m]×1,000[m] の 2 次元平面とした．端末の台数は，テーマパークに来場している歩行者が保持している携帯端末の数として，N(500∼2,000) 台とした．また，シンクは 1 台とし，領域左端および下端から (420[m], 420[m]) の座標に配置した．端末の移動モデルは，テーマパーク内を移動している歩行者を想定し，ランダムウェイポイントモデル [9] 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] 停止するものとした．シンクおよび各端末が備えている通信機能は，参加者が携帯しているスマートフォンやタブレットが備えている機能を想定し，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ものとした．

識別終了時刻 t_{detect} は，クエリ発行時刻から 0.3[s] 後とした．この識別終了時刻までに境界端末の識別処理が終了するように，式 (2.1)，(2.2) を用いてタイマや待ち時間を算出する際の定数 W_q^{max}，$T_{boundary}^{max}$ を，それぞれ 0.1[s]，0.05[s] に設定した．また，データ収集時にパケット衝突が発生しにくくするため，式 (2.3) を用いて待ち時間を算出する際の定数 W_g^{max} を，0.2[s] に設定した．さらに，転送木におけるシンクからの実際の最大ホップ数が，h_{max} にならないようにするために，h_{max} を 12 に設定した．また，式 (2.3) において，パケット衝突を回避するための乱数 *rand* は，[0, 0.5] の範囲内でランダムに決定するものとした．

図 2.10 に，本実験で用いたイベントの形状を示す．このイベントは，一辺が L[m] の正方形である．簡単化のため，イベントの形状を正方形としたが，この形状が複雑になったとしても，境界線が長くなり，境界端末の数が増加するだけであるため，本性能評価と同じ傾向の結果が得られることを実験により確認している．

2.5.2 比較手法

提案手法では，センサデータの受信を用いた部分的なボロノイ図の作成と，センサデータを送信するまでのタイマの設定により，事前にドロネー図を構築することなく，境界端末を識別している．実験では，この動作によるトラヒック削減の

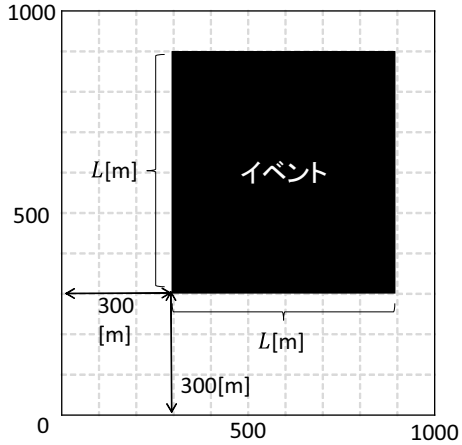


図 2.10: シミュレーション実験で用いるイベント

効果を検証するために，比較対象として，以下の二つの手法の性能を調べた．これらの手法は，提案手法と同じ手順でクエリ伝播とデータ収集を行う．

単純手法

2.1 節で述べた，端末の位置が固定されている環境を想定した既存手法を，モバイルセンサネットワークに適用した手法を，比較対象として用いた．この手法では，クエリを受信した端末から順に，パケット衝突を防ぐためにランダムな待ち時間を設定し，無線通信範囲内に一度センサデータを送信する．これにより，すべての端末がセンサデータを自身の通信範囲内に位置する端末へ送信する．一方，各端末は，自身の隣接端末の情報を把握していないため，いつ自身のすべての隣接端末からセンサデータを受信し終えるのかがわからない．そこで各端末は，2.5.1 項で述べた識別終了時刻において，それまでに隣接端末から受信したすべてのセンサデータを使って，ドロネー図を構築する．このドロネー図を基に自身が境界端末であると判断した端末のみが，自身のセンサデータをシンクへ送信する．

ドロネー図一部構築

2.4.2 項で述べたクエリ伝播により，各端末は通信範囲内に境界線が存在するか否かを判断できるため，通信範囲内に境界線が存在すると判断した端末のみがセンサデータを送信することで，上記の単純手法においてドロネー図を構築する際のトラヒックを削減できるものと考えられる．そこで，もう一つの比較対象として，境界端末を識別する前に，通信範囲内に境界線が存在すると判断した端末，およびその周囲の端末のみを頂点とするドロネー図を構築する手法を用いた．この手法では，クエリ伝播後，通信範囲内に境界線が存在すると判断した端末のみが，パケット衝突を防ぐためにランダムな待ち時間を設定し，無線通信範囲内にセンサデータを送信する．そのセンサデータを受信した端末のうち，観測値の分類が異なる観測値を含んだものを受信した端末も，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ため，同様にランダムな待ち時間を設定し，自身のセンサデータを送信する．この処理を繰り返すことにより，自身の通信範囲内に観測値の分類が異なる端末が存在するすべての端末がセンサデータを送信する．また，この手法においても，各端末は，いつ自身の隣接端末からセンサデータを受信し終えるのかがわからないため，2.5.1 項で述べた識別終了時刻において，それまでに隣接端末から受信したすべてのセンサデータを使って，ドロネー図を構築する．このドロネー図を基に自身が境界端末であると判断した端末のみが，自身のセンサデータをシンクへ送信する．

2.5.3 パラメータ

本実験で変更したパラメータを表 2.2 に示す．これらのパラメータは，基本的にはデフォルト値に設定されるが，そのパラメータの影響を調べる際には，括弧内の範囲で変化させた．

提案手法では，センサデータの受信とセンサデータの送信順序の制御により，事前にドロネー図を構築することなく，境界端末を識別する．この動作によるトラヒックの削減量は，比較手法において，事前にドロネー図を構築する端末の数によって変化するものと考えられる．そこで本実験では，端末数 N を変化させて，

表 2.2: パラメータ設定

パラメータ (意味)	デフォルト値 (値域)
N (端末数)	1,000 (500∼2,000)
L (境界線の一辺の長さ)	600 (200∼600)[m]

表 2.3: メッセージサイズ

手法	手順	メッセージ名	サイズ [B]
提案および ドロネー図一部構築	クエリ伝播	クエリ	78
単純	クエリ伝播	クエリ	70
比較	ドロネー図構築	構築パケット	3 + 24
共通	境界端末識別	データパケット	6 + 24
	データ収集	返信パケット ACK	16 + 24×（センサデータ数） 6

各手法の性能を調べた．

また，ドロネー図を一部構築する手法において，事前にドロネー図を構築する端末の数は，境界線の長さによっても変化する．そこで本実験では，境界線の一辺の長さ L[m] を変化させて，各手法の性能を調べた．

2.5.4 評価指標

- **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の面積を，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センサ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センサ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．

- **総トラヒック**：シンクおよびすべての端末が送信したパケットのアプリケーション層でのサイズの合計．各メッセージサイズを表 2.3 に示す．

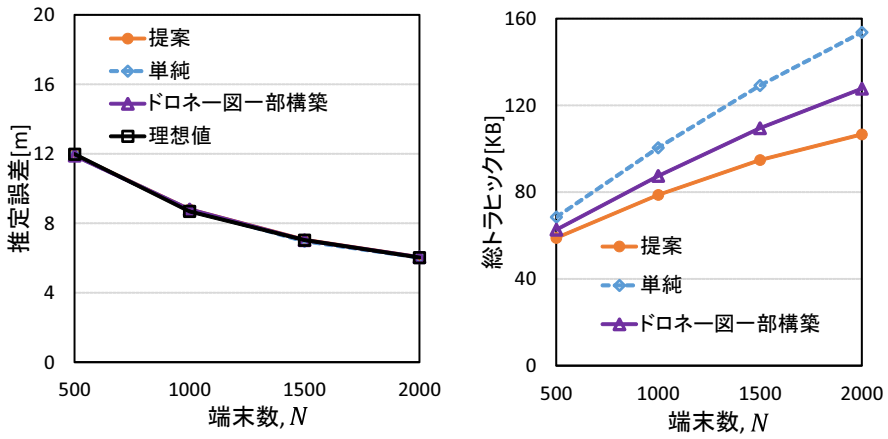
2.5.5 端末数による影響

端末数 N を変化させたときの結果を，図 2.11 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 2.11(a) は推定誤差，図 2.11(b) は総トラヒックを表す．

図 2.11(a) より，各手法とも，推定誤差が理想値と同等の値を示していることがわかる．さらに，端末数の増加に伴い，推定誤差が低下していることがわかる．これは，2.2.3 項で述べたとおり，端末数の増加に伴い，互いに観測値の分類が異なる境界端末間の距離が短くなり，推定される境界線がより実際の境界線に近づくためである．

図 2.11(b) より，各手法ともに，端末数の増加に伴い，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，境界端末の数が増加するためである．また，ドロネー図を一部でのみ構築する手法のトラヒックが，単純手法より小さくなることがわかる．これは，ドロネー図を一部構築する手法では，クエリ伝播時の境界線の存在の認識を利活用することで，境界線から遠い位置における端末によるドロネー図の構築が行われないためである．一方，これらの手法より，提案手法は小さいトラヒックを示していることがわかる．これは，提案手法では，事前にドロネー図を構築することなく，境界端末を識別するためである．さらに，提案手法と比較手法とのトラヒックの差が，端末数の増加とともに増大することがわかる．提案手法では，境界端末のみがセンサデータを送信するように各端末の送信を制御するため，境界端末ではない端末の数が増加するほど，トラヒックの削減量が増加する．

上記を確認するために，各手法における境界端末の識別に要したトラヒックのみを個別に計測した．この値は，提案手法においては 2.4.3 項で述べた境界端末識別処理に要したトラヒックであり，比較手法においては，クエリ伝播後にドロネー図を構築するために端末間でセンサデータを交換する際に要したトラヒックである．また，端末数 2,000 台における総トラヒックの内訳を，クエリ伝播，境界端末識別，およびデータ収集の 3 種に分けて調査した．結果を図 2.12 および図 2.13 に示す．図 2.12 の横軸は端末数 N を表し，縦軸は境界端末の識別に要したトラヒックを示す．図 2.12 より，比較手法において境界端末の識別のために要するトラヒックは，端末数の増加に伴って大きく増大することがわかる．これに対し提案手法



(a) 推定誤差　　(b) 総トラヒック

図 2.11: 端末数による影響

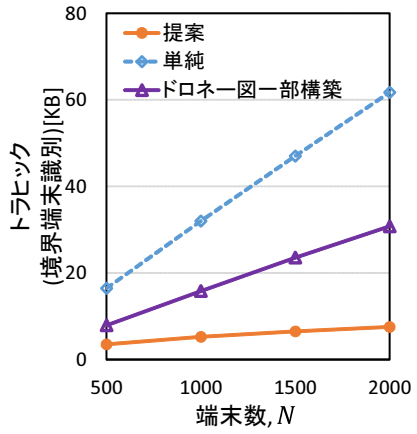


図 2.12: 境界端末の識別に要したトラヒック

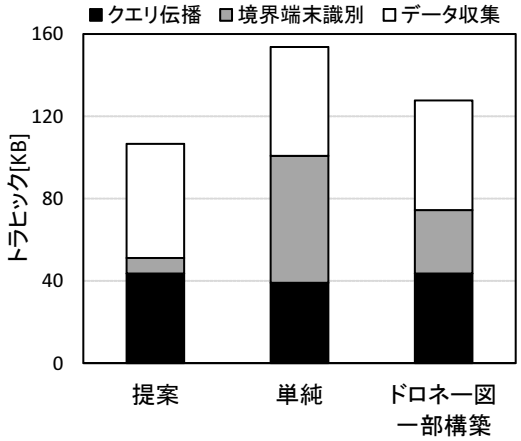
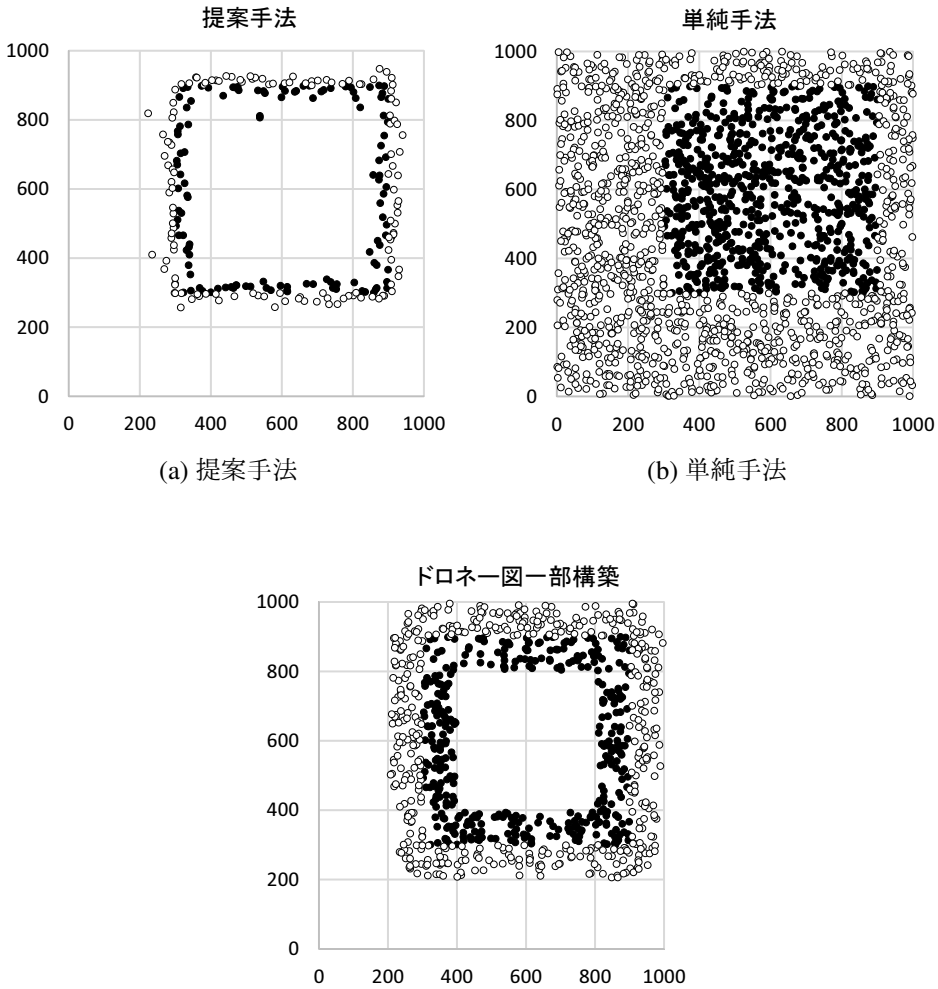


図 2.13: 総トラヒックの内訳

は，事前にドロネー図を構築せずに，境界端末のみがセンサデータを送信する制御を行うことで，境界端末の識別に要するトラヒックを大幅に削減していることがわかる．また前述のとおり，端末数の増加に伴って，その削減量が増加していることがわかる．図 2.13 より，提案手法は，境界端末の識別に要するトラヒックを大幅に削減することによって，境界線検出に要するトラヒックを削減していることがわかる．



(a) 提案手法

(b) 単純手法

(c) 単純手法 (ドロネー図一部構築)

図 2.14: 境界端末識別時にセンサデータを送信した端末の分布

境界端末の識別に要したトラヒックの差異をより詳細に確認するために，端末数が 2,000 台のときに，各手法において，境界端末の識別時にセンサデータを送信した端末の 2 次元平面上における分布を調査した．結果を図 2.14 に示す．図 2.14(a) より，提案手法によりセンサデータを送信した端末は，境界線付近に位置する端末のみであることがわかる．ここで，境界線から遠く離れた端末も数台センサデータを送信していることがわかる．これは，パケット衝突により，境界端末が送信したセンサデータを受信できず，自身が境界端末であると誤って判断してしまった

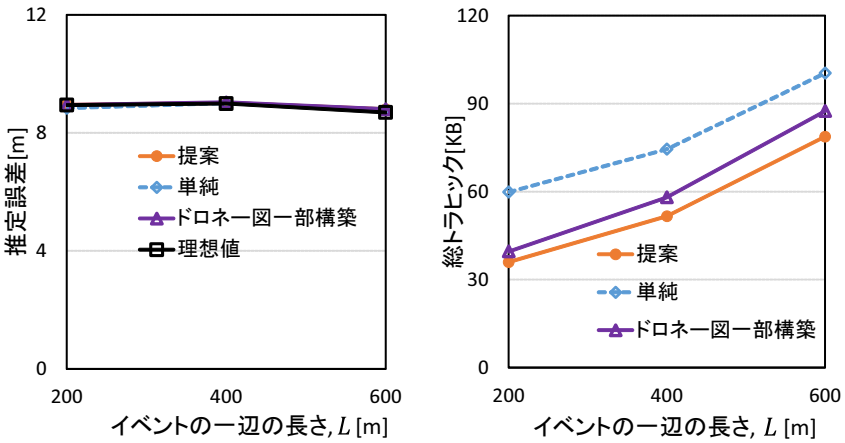
ためである．一方，単純手法を用いた場合，ドロネー図構築のためにすべての端末がセンサデータを送信するため，図 2.14(b) に示すとおり，観測領域全体においてセンサデータが送信される．ドロネー図を一部構築する手法を用いた場合，通信範囲内に境界線が存在しない端末によるセンサデータ送信を抑制できるものの，図 2.14(c) に示すとおり，境界線の周辺において，非常に多くの端末がセンサデータを送信していることがわかる．

2.5.6　境界線の長さによる影響

境界線の一辺の長さ L を変化させたときの結果を，図 2.15 に示す．グラフの横軸は境界線の一辺の長さ L を表し，縦軸はそれぞれ，図 2.15(a) は推定誤差，図 2.15(b) は総トラヒックを表す．

図 2.15(a) より，各手法とも，推定誤差が理想値と同等の値を示していることがわかる．さらに，境界線の長さに対して，推定誤差がほぼ一定であることがわかる．これは，境界線が長くなるほど境界端末は増加するが，観測領域内の端末密度は一定であり，観測値が閾値以上の境界端末と閾値未満の境界端末との間の距離が変わらないためである．

図 2.15(b) より，各手法とも，境界線が長くなるほどトラヒックが増加することがわかる．これは，境界線が長くなるほど，境界端末の数も増加するためである．また，提案手法のトラヒックが，比較手法より小さいことがわかる．これは，2.5.5 項で述べた理由と同じである．また，境界線が長くなるほど，ドロネー図を一部構築する手法と単純手法とのトラヒックの差が小さくなることがわかる．境界線が長くなるほど，ドロネー図を一部構築する手法では，通信範囲内に観測値の分類が異なる端末が存在する端末の数が増えるため，ドロネー図を構築するためにセンサデータを送信する端末の数が増加してしまう．そのため，すべての端末が一度センサデータを送信する単純手法とのトラヒックのが差が小さくなる．一方，提案手法と単純手法のトラヒックの差がほぼ一定であることがわかる．単純手法では，すべての端末がドロネー図構築のためにセンサデータを送信した後，自身が境界端末であると判断した端末がもう一度センサデータを送信する．これに対



(a) 推定誤差 (b) 総トラヒック

図 2.15: 境界線の長さによる影響

し提案手法では，境界端末であると判断した端末が，一度だけセンサデータを送信する．つまり，提案手法と単純手法のトラヒックの差は，すべての端末が一度センサデータを送信するために要するトラヒックである．これは境界線の長さに関わらず一定であるため，境界線が長くなっても，単純手法に対するトラヒック削減量が変わらない．

2.6 むすび

本章では，モバイルセンサネットワークにおけるダイアグラムの逐次作成を用いた境界線検出手法を提案した．提案手法では，受信したセンサデータを基に各端末がボロノイ図を逐次作成し，自身が境界端末か否かを判断する．そして，境界端末と判断した端末のみが，自身と観測値の分類が異なる端末に近いほど短くなるタイマを設定してセンサデータを送信することで，境界端末のみがセンサデータを送信するように送信順序を制御する．これにより，事前にドロネー図を構築することなく，境界端末を識別できる．

シミュレーション実験により，提案手法は，事前にドロネー図を構築する手法と比較して，推定誤差を維持しつつ，トラヒックを削減できることを確認した．さら

に，端末数が増加するほど，トラヒックの削減量が増加することを確認した．モバイルセンサネットワーク内のトラヒックを大きく削減することにより，ネットワークに参加している端末のバッテリ消費の低減が期待できる．また，無線通信帯域に余裕が生まれるため，他のアプリケーションが同一のネットワークを使用できるようになる．

一方，2.4.5 項で述べたように，提案手法は収集するセンサデータ数を制御できないため，境界端末数が極端に多い環境では，境界端末のみからセンサデータを収集するとはいえ，大量のトラヒックが発生してしまうという問題がある．上記のような，収集データ数を制御できないことによって発生する問題については，第4章で取り組む．また，本章ではクエリ発行時刻における境界線の検出を想定しており，定期的な境界線検出を行う場合は考慮していない．定期的な境界線検出を行うために提案手法を用いた場合，クエリ伝播を定期的に行う必要があり，トラヒックが増大してしまうという問題がある．この問題を解決するためには，1 回目のクエリ伝搬以降は，クエリの伝播なしで境界線を検出できる方法が求められる．第3章では，この問題について取り組む．

また，2.4.5 項で述べたとおり，環境によっては，すべての端末から個別にセンサデータを収集する手法のほうが，提案手法より早くデータ収集を完了できる場合があるものと考えられる．そこで，端末数や通信速度を変化させながら，さらなる実験を行い，データ収集が完了するまでの時間の観点から，提案手法の適用範囲を調べる予定である．

第3章　観測値の時間的変化の監視を用いた定期的な境界線検出手法

3.1　まえがき

第2章では，ある時刻におけるイベントの位置と形状を要求するアプリケーションを想定し，低トラヒックでイベントの境界線を検出するために，ダイアグラムの逐次作成を用いた境界線検出手法を提案した．一方，イベントが時間経過に伴って変化する場合，その変化を確認するために，定期的にイベントの位置と形状を要求するアプリケーションも存在する．この要求を満足するためには，イベントの境界線を定期的に推定する必要がある．このように，時間経過に伴うイベントの変化を確認するために，定期的に境界線を推定することを，以降では「境界線のモニタリング」と表記する．境界線のモニタリングを行うために，第2章で提案した手法をそのまま用いると，1.3.2項で述べたとおり，定期的にクエリのフラッディングを行う必要があり，端末数が多い環境では，そのためのトラヒックが著しく増大する．第2章で提案した手法におけるクエリのフラッディングは，観測領域内に存在するすべての端末にアプリケーション要求を通知するために行っている．しかし，定期的に境界線検出を行う場合，各端末に1度だけアプリケーション要求を通知すれば，各端末は既に要求を把握している状態になるため，その後新たに端末が参加しない場合，これを定期的に行う必要はない．一方，クエリのフラッディングは，各端末に境界線の存在を認識させ，境界端末の識別処理を開始するため，およびセンサデータをシンクへ送信するための転送木を構築するためにも行っている．つまり，クエリのフラッディングを行わない場合，以下の二

つの問題が発生する．

1. 各端末がどのようにして自身の通信範囲内に境界線が存在するか否かを確認するか．

2. 転送木を構築することなく，どのようにして境界端末のセンサデータを収集するか．

問題 1 を解決するためには，1.3.2 項で述べたとおり，各端末における観測値の時間的変化を用いることが有効である．本章では，これを利用し，クエリをフラッディングすることなく，境界端末を定期的に識別できる手法を提案する．提案手法では，定期的に監視した観測値の時間的変化を基に，境界線を跨いでからの経過時間が短いと判断した端末のみがセンサデータを送信し，そのセンサデータを受信した端末から順に，境界端末の識別処理を行う．これにより，クエリをフラッディングすることなく，定期的に境界端末を識別できる．さらに，問題 2 を解決するために，事前に経路を構築することなく，目標位置に最も近い端末へメッセージを送信可能なジオルーティングプロトコル [30] を用いて，上記の方法で識別した境界端末のセンサデータをシンクへ送信する．

以下では，まず 3.2 節で想定環境について述べ，3.3 節で関連研究について述べる．その後，3.4 節で提案手法について述べた後，3.5 節でシミュレーション実験の結果を示す．最後に 3.6 節で本章のまとめと課題について述べる．

3.2　想定環境

端末構成および境界線推定方法は，基本的には 2.2 節で述べたものと同じであるが，各端末は，2.4.2 項で述べたクエリのフラッディングなどにより，アプリケーションの要求を事前に把握しており，その要求に従って動作するものとする．具体的には，各端末は，シンクの位置，観測時刻，観測値の種別，イベント判定のための観測値の閾値を所持しており，任意の時刻において，要求に合致する種別の現象をセンシングできるものとする．また，本章で想定するアプリケーションは，イベントの位置と形状を観測周期 C[s] で要求する．シンクはアプリケーショ

ンに従い，観測時刻 T_i（$= T_0 + i \cdot C; i =$1，2，$\cdots$）[s] ごとに，センサデータを収集し，境界線を推定する．

また本研究では，端末の電源 On/Off や，端末の観測領域内へ流入や観測領域からの流出などによる，新たな端末のネットワークへの参加や，端末のネットワークからの離脱は想定していない．しかし，アプリケーションが一定期間境界線をモニタリングする場合，その期間中に上記のような端末の参加や離脱が発生することは十分に考えられる．このような環境において，本研究では想定していない状況が発生するのは，端末が新たにネットワークに参加する場合である．この場合，その端末はアプリケーション要求を把握していないため，その要求に従って動作できない．

しかし，観測時刻間に新たに参加した端末が少ない場合であれば，当該端末が隣接端末からセンサデータなどのメッセージを受信したときに，そのメッセージを送信した端末へアプリケーション要求を問い合わせすることで，容易に要求を把握できる．一方，観測時刻間に大量の端末が新たに参加し，どの隣接端末も要求を把握していないという状況が発生する場合，上記の方法では要求を把握できない端末が現れる．この場合，上記の方法で要求を把握できた端末が，2.4.2 項で述べたクエリ伝播を，まだ要求を把握していない端末に対してのみ行うことで，シンクからもう一度クエリを伝播するより少ないトラヒックで，新たに参加した各端末に要求を把握させることができる．このように，端末の参加が発生する環境においても，提案手法は容易に適用できるものと考えられる．

3.3 関連研究

本節では，固定センサネットワークにおけるイベントの境界線や等値線のモニタリングに関する既存研究について述べる．また，モバイルアドホックネットワークにおけるネットワークの境界線のモニタリングに関する既存研究についても述べる．

3.3.1 固定センサネットワークにおけるイベントの境界線のモニタリング

文献 [7, 38, 39, 42, 64, 65, 66, 82, 88, 92] では，時間経過に伴ってイベントの境界線が変化した場合，観測値が時間的に変化した端末のみの情報を用いて，低トラヒックで境界線をモニタリングする手法を提案している．文献 [34, 38, 39, 66, 88, 92] では，観測値の分類が異なる隣接端末が存在する端末を，境界線付近の端末と定義している．文献 [34, 39, 88] で提案されている手法では，シンクがすべての端末の位置情報と端末識別子を把握している環境において，観測値が時間的に変化した境界線付近の端末は，その観測値と端末識別子を，その他の境界線付近の端末は端末識別子のみをシンクへ送信する．文献 [92] で提案されている手法では，自身が境界線付近の端末であると判断した端末間で構築したネットワーク自体をイベントの境界線としている．境界線の変化によって，境界線付近の端末が変化した場合，境界線から離れた端末をネットワークから除外し，新たな境界線付近の端末をネットワークに追加することで，境界線のモニタリングを実現している．文献 [38, 66] で提案されている手法では，今回の観測時刻における境界線付近の端末が，次回の観測時刻における境界線付近の端末が存在する可能性がある範囲を予測し，その付近でのみ境界線付近の端末を探索する．また，文献 [38] で提案している手法では，境界線付近の端末どうしを辺で結んだ閉路グラフを構築し，そのグラフ自体をイベントの境界線としている．境界線の変化によって，境界線付近の端末が変化した場合，文献 [92] で提案されている手法と同様の処理によって，グラフを部分的に更新することで，境界線のモニタリングを実現している．

文献 [82] で提案されている手法では，2.2.3 項で述べたドロネー図をはじめとする三角形分割において，観測値が閾値以上の端末と隣接する端末のみで構成される三角形分割の境界線を，イベントの境界線としている．境界線の変化によって，端末の観測値が時間的に変化した場合，観測値が閾値未満の端末のみで構成される三角形を除外し，観測値が閾値以上に変化した端末を含む三角形を追加することで，境界線のモニタリングを実現している．

文献 [7] で提案されている手法では，観測値が閾値以上の端末を包含する χ-

shape[19] を，イベントの境界線としている．この手法では，シンクが観測値が閾値以上の端末からセンサデータを収集し，そのセンサデータを基に χ-shape を作成することによって，境界線を推定する．また，境界線の変化によって，観測値が時間的に変化した端末のみがセンサデータをシンクへ送信し，シンクが χ-shape を更新することによって，境界線をモニタリングする．

文献 [42, 64, 65] では，観測領域が複数のクラスタに分割されており，クラスタヘッドが自身のクラスタ内の端末の位置情報と端末識別子を把握している環境において，各クラスタヘッドが自身のクラスタ内における境界線の変化を確認することで，境界線全体をモニタリングする手法を提案している．この手法では，観測値が閾値以上の端末が，観測値と端末識別子を自身が所属するクラスタのクラスタヘッドへ送信し，各クラスタヘッドがその情報を基に，自身のクラスタ内の境界線を推定する．境界線の変化によって，観測値が時間的に変化した端末のみが，観測値と端末識別子を自身が所属するクラスタのクラスタヘッドへ送信する．クラスタヘッドは，受信した情報を基に，境界線が変化したと判断した場合にのみ，自身のクラスタ内の境界線の情報をシンクへ送信する．

上記の研究は，境界線のモニタリングに要するトラヒックの削減を目的としている点や，観測値の時間的変化を用いている点が本研究と類似している．一方，これらの研究は，端末の位置が固定されているため，境界線が変化しない限り，前回の観測時刻における境界線付近の端末が，今回の観測時刻においても，境界線付近の端末であることが保証されている環境を想定している点が本研究と異なる．これらの既存研究とは異なり，本研究では，各端末が自由に移動する環境を想定しており，境界線が変化しない場合においても，境界線付近の端末が変化してしまうため，既存研究のアプローチをそのまま適用できない．

3.3.2 固定センサネットワークにおける等値線のモニタリング

文献 [61, 89] では，観測領域内に存在するすべての端末のセンサデータを用いて等値線を推定する環境において，等値線をモニタリングするために，観測値の時空間的相関性を用いて，収集するセンサデータ数を削減する手法を提案している．

これらの手法では，各端末は，自身の観測値が前回の観測時刻に取得したものからの許容誤差範囲内にある，または自身の隣接端末が送信した観測値からの許容誤差範囲内にある場合，自身の観測値の送信を中止する．またシンクは，受信した観測値を用いて，送信されなかった観測値を補間し，等値線図を作成する．

これらの研究は，等値線のモニタリングに要するトラヒックの削減を目的としている点が本研究と類似している．一方，端末の移動を考慮していない点，およびすべての端末のセンサデータを用いて等値線を推定することを想定している点が本研究と異なる．

3.3.3 固定センサネットワークにおけるイベントトに関する情報の定期的な把握

文献 [4] では，イベントの境界線以外の情報（位置，大きさ，移動速度など）を定期的に要求するアプリケーションを対象として，ユーザの要求に応じて，収集する情報を変更できる手法を提案している．この手法では，観測値が閾値以上の端末を根および内部ノード，観測値の分類が異なる隣接端末が存在する端末を葉ノードとする木構造を構築する．この木構造の根にあたる端末が，自身の子孫からセンサデータを収集し，ユーザの要求に応じて集約することで，ユーザの要求に応じたイベントの情報のみを取得できる．また，境界線の変化に応じて，葉ノードに該当する端末を追加，削除し，木構造を更新することで，ユーザの要求に応じたイベントの情報の定期的な把握を低トラヒックで実現している．

この研究は，イベントの情報の定期的な把握に要するトラヒック削減を目的としている点が本研究と類似している．一方，端末の移動を考慮していない点，およびイベントの形状がアプリケーション要求の対象となっていない点が本研究と異なる．

3.3.4 モバイルアドホックネットワークにおけるネットワークの境界線のモニタリング

文献 [43] では，モバイルアドホックネットワークにおいて，すべての端末を包含する凸包をネットワークの境界線として，端末の移動によって凸包が変化した場合，部分的に凸包を修正する手法を提案している．この研究は，アプリケーションが凸包に対してマージンエリアを設定して，その変化をある程度許容する環境を想定しており，凸包がマージンエリアから出た場合にのみ，凸包の更新を要求する．この手法では，[44, 45] で提案されている手法によって検出した凸包の各頂点に対して，凸包がマージンエリア内に留まっていることが保証される領域（Safe zone）を設定し，頂点に位置する端末が Safe zone を出た場合にのみ，部分的に凸包を更新する．

この研究は，境界線のモニタリングに要するトラヒックの削減を目的としている点や，端末の移動を考慮している点が本研究と類似している．一方，2.3.4 項で述べたとおり，境界線付近の端末の定義が本研究のものと異なるため，この既存研究のアプローチは適用できない．

3.4 提案手法

本節では，まず提案手法の概要を述べ，その後，詳細な手順について説明する．

3.4.1 動作概要

提案手法では，時刻 T_0，すなわち初回の観測時刻において，第 2 章で提案した手法により，境界線検出を行う．その後，前回の観測時刻 T_{i-1}（i =1，2，$\cdots$）から次回の観測時刻 T_i までの間において，3.4.2 項で述べる方法で，各端末が自身の観測値の時間的変化を監視することで境界線を跨いだか否かを確認し，観測時刻 T_i において，観測時刻付近で跨いだと判断した端末のみが自身のセンサデータを送信する．このセンサデータを受信した端末から順に，3.4.3 項で述べる方法で，

自身が境界端末か否かを判断する．最後に，境界端末が送信したセンサデータを，3.4.4 項で述べる方法でシンクへ送信する．

3.4.2　境界線の存在の認識

各端末は，アプリケーションが設定する観測周期 C[s] を k（=1，2，$\cdots$）等分した周期 c（$= \frac{1}{k} \cdot C$）[s] で，時刻 t_j（$= T_{i-1} + j \cdot c; j$ =1，2，$\cdots$）[s] にセンシングを行う．時刻 t_j の観測値が t_{j-1} の観測値と分類が異なる場合，境界線を跨いだ時刻 t_b として t_j を記録する．例えば図 3.1 において，ある端末が T_{i-1} から 6 回目の観測で，5 回目までに観測した値とは分類が異なる値を観測している．この端末は，6 回目に観測した時点で自身が境界線を跨いでいるものとし，6 回目に観測した時刻を t_b として記録する．

観測時刻 T_i では，すべての端末がセンシングおよび自身の位置情報の取得を行い，センサデータを生成する．また，T_{i-1} から T_i までの間に境界線を跨いだ端末は，記録しておいた t_b を用いて，自身の近くに境界線が存在する可能性があるか否かを判断する．ここで，t_b が T_i に近いほど，T_i において，その端末の近くに境界線が存在する可能性が高い．そこで各端末は，t_b が以下の条件を満たす場合にのみ，自身の近くに境界線が存在する可能性があると判断する．

$$T_i - t_b \leq s \, (c \leq s \leq C). \tag{3.1}$$

ここで，s は，t_b が T_i に近いか否かを判断するためのパラメータであり，全端末共通で予め設定される値である．

式 (3.1) に従い，自身の近くに境界線が存在する可能性があると判断した端末は，ランダムな待ち時間を設定して，自身の通信範囲内へセンサデータを送信する．このセンサデータを受信した端末のうち，センサデータの送信元端末と観測値の分類が異なる端末は，自身の通信範囲内に境界線が存在すると認識できるため，これらの端末から順に，2.4.3 項で述べた境界端末識別処理を開始できる．ここで，式 (3.1) を満たしたすべての端末がセンサデータを送信しなくても，2.4.3 項で述べた境界端末識別処理は実行できる．これは，識別処理中に境界端末であると判断し

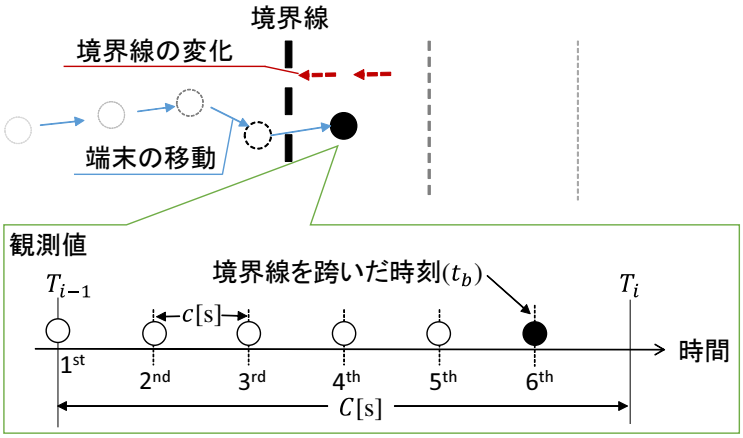


図 3.1: 観測値の時間的変化の監視の例

た端末が送信したセンサデータが，式 (3.1) を満たした端末が送信するセンサデータの代わりになるためである．そこで，式 (3.1) を満たした一部の端末のみにセンサデータを送信させるために，式 (3.1) を満たした端末が，本項の処理によって送信されたセンサデータを待ち時間内に受信した場合，自身のセンサデータの送信を中止し，次項で述べる境界端末識別処理に移行する．

3.4.3　境界端末の識別

前項または本項の処理によって送信されたセンサデータのうち，観測値の分類が異なるものを受信した各端末は，2.4.3 項で述べた手順に従って，境界端末を識別する．ここで，第 2 章で提案した手法とは異なり，提案手法はクエリのフラッディングを行わないため，転送木が構築されない．そのため，境界端末がセンサデータを送信する際は，2.4.2 項で述べた転送木上の親に該当する端末の識別子は付与せずに，自身の通信範囲内へ送信する．このセンサデータは，送信元端末の全隣接端末が受信・保持する．

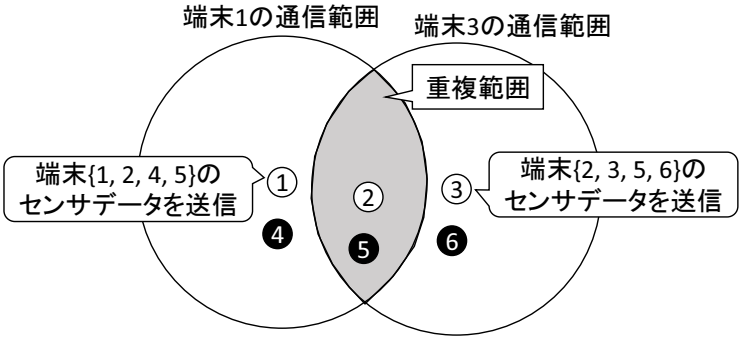


図 3.2: 送信されるセンサデータの重複（端末 $\{1, \cdots, 6\}$ はすべて境界端末）

3.4.4　データ収集

上記の処理によって自身が境界端末であると判断した各端末は，センサデータをシンクへ送信する．そのための単純な方法として，各境界端末が独立して，自身のセンサデータのみを含むメッセージをシンクへ送信する方法が考えられる．しかし，この方法を用いると，特に境界端末が多数存在する場合，データ収集時に発生するメッセージ数が著しく増大してしまう．

ここで境界端末は，3.4.3 項で述べた識別処理中に隣接端末から受信したセンサデータを所持している．これらのセンサデータは，すべて境界端末と判断した端末のものであるため，これらを一つのパケットにまとめて送信することで，データ収集に要するメッセージ数の増大を抑制できる．そこで提案手法では，3.4.3 項で送信された全センサデータを，以下で述べる方法に従い，最小限の端末を用いてシンクへ送信する．具体的には，文献 [34, 88] で提案されている手法と同様の処理によって，境界端末の中から，シンクへセンサデータを送信する端末（代表端末）を決定する．各代表端末は，後述するジオルーティングを用いて，自身と隣接する境界端末から受信したセンサデータを一つのパケットにまとめて，シンクへ送信する．以下では，その詳細について説明する．

代表端末の決定

データ収集に要するメッセージ数を抑えるためには，シンクが 3.4.3 項で送信された全センサデータを受信できるという条件のもと，代表端末の数を可能な限り少なくすればよい．ここで，図 3.2 における端末 1 が代表端末となり，隣接する境界端末のセンサデータを一つのパケットにまとめてシンクへ送信した場合，境界端末 $\{1, 2, 4, 5\}$ のセンサデータが送信される．そのため，端末 $\{2, 4, 5\}$ がセンサデータ送信を中止することで，代表端末数を削減できる．このとき，端末 1 の通信範囲外に位置する境界端末 3 または 6 は，代表端末として自身のセンサデータをシンクに送信する必要があるが，例えば端末 3 が代表端末となり，境界端末 $\{2, 3, 5, 6\}$ のセンサデータをまとめて送信した場合，端末 1 と 3 で通信範囲が重複している領域（重複範囲）に位置する端末 $\{2, 5\}$ のセンサデータが重複してシンクに送信され，データ収集時に無駄なトラヒックが発生してしまう．ここで，図 3.2 の端末 3 が，端末 1 が代表端末となったこと，およびその位置を認識できれば，シンクへ送信するセンサデータから重複範囲内のものを除くことができ，トラヒックの増大を抑制できる．提案手法では，代表端末数を削減しつつ，重複してセンサデータが送信されることを防ぐため，データ収集に先立ち，以下の手順に従い，代表端末および送信するセンサデータを決定する．

1. 3.4.3 項における境界端末識別終了後，自身が境界端末であると判断した端末は，自身のセンサデータおよび受信したセンサデータを，シンクへ送信するセンサデータのリスト（送信データリスト）に登録する．また，ランダムに設定した待ち時間の後，自身の位置情報を含む立候補メッセージを自身の通信範囲内へ送信し，代表端末となる．待ち時間内に立候補メッセージを受信した端末は，隣接端末が代表端末となったことを認識し，自身のメッセージ送信を中止する．これにより，互いに通信範囲内に存在しない端末群が代表端末となる．また，最初に受信した立候補メッセージの送信元端末（便宜上，n_{first} と表記する）の位置情報を記録する．

2. 立候補メッセージを二つ受信した場合，その端末は重複範囲内に位置する（例えば，図 3.2 における端末 2）．そこでこの端末は，代表端末によるセンサ

データの重複送信を防ぐため，ランダムに設定した待ち時間の後，n_{first} の位置情報を付与した重複回避メッセージを，自身の通信範囲内へ送信する．このメッセージを受信した端末は，メッセージに含まれる位置情報から，n_{first} が所持するセンサデータが含まれる範囲（半径 r[m] の円）を認識する．また，待ち時間内に他の端末が送信した重複回避メッセージを受信した場合，既に n_{first} の位置が通知されたことを認識し，自身の重複回避メッセージ送信を中止する．

3. 代表端末は，受信した重複回避メッセージに含まれる位置情報が自身の位置と異なる場合，この位置情報を基に，自身の通信範囲との重複範囲を算出し，重複範囲内のセンサデータを，送信データリストから削除する．これにより，代表端末によるセンサデータの重複送信を防ぐことができる．例えば，図3.2において，端末2が端末1の位置情報が含まれた重複回避メッセージを送信したものとする．この重複回避メッセージを受信した端末3は，代表端末となった端末1が端末 $\{2, 5\}$ のセンサデータを所持していることを認識し，自身の送信データリストから，端末 $\{2, 5\}$ のセンサデータを削除する．ここで，境界端末が三つ以上の立候補メッセージを受信した場合，その境界端末は重複回避メッセージを送信しない．これは，二つの立候補メッセージを受信したときに送信した重複回避メッセージによって，その境界端末の通信範囲内にいる端末が，既に n_{first} の位置を把握しているためである．

上記の手順2と手順3では，重複してセンサデータが送信されることを防ぐため，一部の代表端末が受信したセンサデータを削除する．しかし，この処理を行っても，以下の定理より，シンクは識別処理中に送信されたすべてのセンサデータを受信できる．

定理 1. 手順2および手順3を行った後，識別処理中に送信されたセンサデータは，いずれかの代表端末が必ず所持している．

証明. 手順2および手順3を行った後，どの代表端末も所持していないセンサデータ sd_{miss} が存在すると仮定する．ここで，sd_{miss} を送信した境界端末を n_{miss} と表記し，n_{miss} の通信範囲内に位置する境界端末の集合を N_r と表記する．N_r に含ま

れる境界端末は，境界端末識別中に n_{miss} から sd_{miss} を受信している．また，N_r に含まれる境界端末の通信範囲は，少なくとも n_{miss} の位置において重複している．

上記の仮定に従い，N_r に含まれるすべての代表端末が，N_r に含まれない n_{first} の位置情報を含む重複回避メッセージを受信し，sd_{miss} を送信データリストから削除しているものとする．一方，n_{first} の位置情報を基に sd_{miss} が削除されたということは，n_{miss} が n_{first} の通信範囲内に位置していることになる．これは，sd_{miss} を所持した n_{first} が N_r に含まれていることを意味しており，N_r に含まれるすべての代表端末が sd_{miss} を削除している，すなわち，どの代表端末も所持していないセンサデータ sd_{miss} が存在するという仮定と矛盾する．従って，定理 1 は成り立つ．

□

ジオルーティングによるセンサデータ送信

識別終了時刻から τ[s] 後に，代表端末はランダムな待ち時間を設定し，送信データリスト内のセンサデータを一つのパケットにまとめて，文献 [30] で提案された手法を基にしたジオルーティングプロトコルを用いて，シンクへセンサデータを送信する．

まず代表端末が自身の位置情報をパケットヘッダに書き込み，隣接端末へパケットを送信する．パケットを受信した端末は，送信元端末，通信範囲およびシンクの位置から転送エリア（送信元端末よりもシンクの位置に近く，かつその領域内の端末が互いに直接通信可能な領域）を算出する（図 3.3）．自身が転送エリア外に位置している場合は，受信したパケットを破棄する．自身が転送エリア内に位置している場合は，パケットを送信した端末から遠いほど短くなる待ち時間を設定し，自身の位置情報をパケットヘッダに書き込み，隣接端末へパケットを送信する．待ち時間内に転送エリア内の他の端末のパケット送信を検知した場合は，自身のパケット送信を中止する．これにより，パケットは転送されるごとにシンクの位置へと近づいていく．シンクがパケットを受信した場合は，ACK を送信する．ACK を受信した端末は，自身のパケット送信を中止する．これにより，シンクは代表端末が送信したパケットを受信できる．

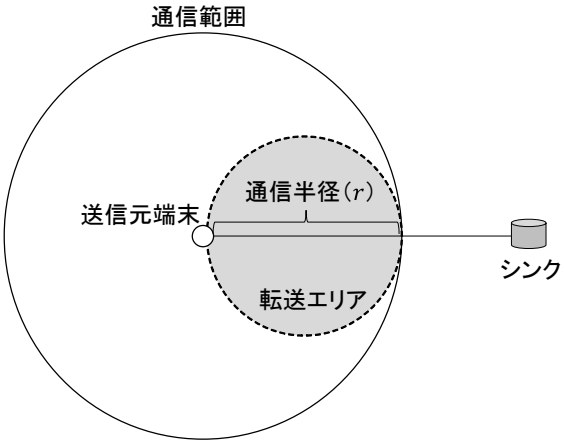


図 3.3: 転送エリア

3.4.5 考察

提案手法では，各端末が観測時刻間に自身の観測値の時間的変化を監視することで，境界線を跨いだか否かを確認する．そして，跨いだ時刻が観測時刻に近い場合，自身の近くに境界線が存在する可能性があると判断し，センサデータを送信する．このセンサデータを受信した端末から順に，2.4.3 項で述べた境界端末の識別処理を実行する．この処理により，すべての境界端末を識別できるか否かは，時刻 $(T_i - s)$ から T_i の間に境界線を跨いだ端末の数，および端末密度に依存する．本項では，時刻 $(T_i - s)$ から T_i の間に境界線を跨いだ端末の数と端末密度の関係について議論する．また，時刻 $(T_i - s)$ から T_i の間に 1 台も境界線を跨がない場合について議論する．さらに本項では，上記の処理によってすべての境界端末を識別するまでの時間と，3.4.2 項の処理によってセンサデータを送信する端末の数の関係についても議論する．

境界線を跨いだ端末の数と端末密度の関係

提案手法により，クエリを定期的にフラッディングすることなくすべての境界端末を識別できるか否かは，時刻 $(T_i - s)$ から T_i の間に境界線を跨いだ端末の数，および端末密度に依存する．例えば，ある単一のイベントの境界線に対して，時刻 $(T_i - s)$ から T_i の間にその境界線を跨ぎ，センサデータを送信した端末が 1 台

であった場合においても，端末密度によっては境界端末をすべて識別できる．具体的には，上記の端末が送信したセンサデータを受信した端末がボロノイ図を作成し，自身が境界端末であると判断する端末が 1 台でも存在すれば，その端末が自身のセンサデータを送信する．このセンサデータに対しても同様の判断を行う端末が 1 台でも存在すれば，その端末が自身のセンサデータを送信する．この処理を繰り返すことにより，当該イベントの境界線に対応したすべての境界端末がセンサデータを送信できるほど端末密度が高い環境であれば，時刻 $(T_i - s)$ から T_i の間にその境界線を跨ぐ端末がたとえ 1 台であっても，境界端末をすべて識別できる．

一方，上記のような連鎖的な識別がすべての境界端末まで伝播しないほど端末密度が低い環境では，式 (3.1) に従ってセンサデータを送信する端末数が非常に少ない場合，すべての境界端末を識別できない可能性がある．しかし，モバイルセンサネットワークを用いてユーザ参加型センシングを行う環境では，1.1 節で述べたとおり，センシングに参加しているユーザにインセンティブを与えることで，多数のユーザからのセンサデータの収集を行っている．このような背景のもと，実際にユーザ参加型センシングを行う環境では，ネットワークを構成する端末の密度は高くなるものと考えられるため，連鎖的な識別が途中で止まる状況は発生しにくい．

時刻 $(T_i - s)$ から T_i の間に境界線を跨いだ端末が 1 台も存在しない場合

時刻 $(T_i - s)$ から T_i の間に境界線を跨いだ端末が 1 台も存在しない場合，3.4.2 項で述べた処理ではセンサデータが一つも送信されないため，境界端末の識別処理が始まらず，境界端末を識別できなくなってしまう．この問題は，以下の状況において発生する．

1. すべての端末が，境界線の移動方向と同じ方向に，同じ速さで移動する．
2. 時刻 $(T_i - s)$ から T_i の間に境界線を跨いだ，または跨ぐはずであったすべての端末が，ネットワークから離脱してしまう．
3. センシング周期 c[s] およびパラメータ s を非常に小さい値に設定する．

4. 観測周期 C[s] を非常に小さい値に設定する．

状況 1 は，端末がモニタリング対象のイベントから逃げるように移動する場合に，発生する可能性がある．具体的には，局所的に豪雨が発生している領域や，PM2.5 などの微粒子濃度が高い領域など，ネットワークを構成する端末を所持する歩行者にとって不快なものがモニタリング対象のイベントである場合，端末がそのイベントから逃げるように移動することが多いものと考えられる．この状況において，当該イベントが全方位に広がっていくようにその形状が変化する場合，端末と境界線が同じ方向に移動する可能性がある．さらに，その速さが同じであれば，状況 1 が発生するが，すべての端末が境界線が同じ速さで移動することは非現実的である．端末と境界線が同じ方向に移動している場合においても，その速さが異なる場合，境界線を跨ぐ端末が少なからず現れるため，その端末から連鎖的に境界端末を識別できる．

状況 2 は，ネットワークを構成する端末数が多い環境では，発生しにくいものと考えられる．これは，ネットワークを構成する端末を所持する歩行者は，各々様々な方向へ移動することが一般的であり，端末数が多い環境では，境界線を跨ぐ端末も多くなるためである．しかし，状況 1 のように，端末と境界線が同じ方向に移動する場合など，端末が境界線を跨ぎにくくなる環境では，状況 2 が発生する可能性は十分にあるものと考えられる．この場合，時刻 $(T_i - s)$ から T_i の間に境界線を跨ぐ端末が存在しなくなるため，観測値の時間的変化の監視を用いずに境界端末を定期的に識別できる方法が必要となる．

状況 3 の場合，各端末は，アプリケーションがモニタリングを要求する期間中，絶えず非常に短い周期でセンシングすることになり，電力を大量に消費してしまう．これは，その端末を所持するユーザがセンシングに参加する意欲を削いでしまう可能性があるため，好ましくない．上記より，モバイルセンサネットワークを用いてユーザ参加型センシングを行う環境において，時刻 $(T_i - s)$ から T_i の間に境界線を跨ぐことが困難になるほど小さい c および s を設定することは，現実的ではない．一方，c および s を非常に小さい値に設定してしまい，端末が 1 台も境界線を跨がず，境界端末を識別できなかった場合，c および s を再設定するためのメッセージをフラッディングすればよい．

状況 4 は，アプリケーションがリアルタイムな境界線のモニタリングを要求する場合に発生する．この場合，観測時刻間に境界線と端末がほとんど移動できない場合が多く，端末が境界線を跨ぎにくくなるため，観測値の時間的変化の監視を用いずに境界端末を定期的に識別できる方法が必要となる．一方，上記が実現できたとしても，ごく短い周期で，識別したすべての境界端末のセンサデータを収集することになるため，大量のトラヒックが発生してしまう．この問題については，5.2.1 項で議論する．

すべての境界端末を識別するまでの時間と，3.4.2 項の処理によってセンサデータを送信する端末数の関係

境界線を跨いだ端末の数と端末密度の関係に対する考察でも述べたとおり，提案手法では，連鎖的な識別がすべての境界端末へ伝播するほど端末密度が高い環境であれば，時刻 $(T_i - s)$ から T_i の間に境界線を跨いだ端末がたとえ 1 台であっても，境界端末を識別できる．しかし，3.4.2 項の処理によってセンサデータを送信する端末が少ない場合，連鎖的な識別が始まる地点が少なくなるため，連鎖的な識別がすべての境界端末に伝播するまでの時間が長くなる．一方，3.4.2 項の処理によってセンサデータを送信する端末が多い場合，連鎖的な識別が始まる地点が多くなり，各地点から同時並列的に境界端末が識別されるため，連鎖的な識別がすべての境界端末に伝播するまでの時間が短くなる．このように，提案手法において，すべての境界端末を識別するまでの時間は，3.4.2 項の処理によってセンサデータを送信する端末数に依存するものと考えられる．

3.5 性能評価

本節では，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，境界線のモニタリングが求められる環境における提案手法の性能を評価するために，テーマパークや花火大会といった歩行者の通行を阻害するものが極めて少ない環境において，テーマパーク事業者や大会主催者が，気温が局所的に高くなっている領域の境界線をモニタリングし，来場者に対して熱中症

予防を注意喚起する状況を想定した．実験では，2.5 節と同じネットワークシミュレータを用いた．

3.5.1　シミュレーション環境

観測領域や端末構成は，基本的には 2.5.1 項で述べたものと同じであるが，シンクは，シミュレーション開始から C[s] 間隔で，シミュレーション終了までデータ収集を行うものとする．識別終了時刻は，各観測時刻から 0.3[s] 後とし，代表端末がセンサデータの送信を開始する時刻は，識別終了時刻から 0.2（$\tau = 0.2$）[s] 後とした．また，観測時刻間に各端末がセンシングを行う周期は 15（$c = 15$）[s] とした．3.4.2 項で述べたパラメータ s は，$\{15, C\}$ の二つを用いた．

図 3.4 に，本実験で用いたイベントの形状を示す．このイベントは，一辺が 300[m] の正方形であり，図の位置を初期位置とし，矢印の方向に 1[m/s] の速度で等速移動する．この移動速度は，気温が局所的に高い領域が時間経過に伴って徐々に変化するときの境界線の速度を想定して設定した．

3.5.2　比較手法

提案手法では，3.4.2 項で述べた通り，各端末が自身の観測値の時間的変化を監視し，境界線を跨いだか否かを確認することで，各端末が境界線の存在を認識するために要する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，第 2 章で提案した手法を，観測時刻ごとに適用する手法を比較対象とした．

第 2 章で提案した手法では，クエリのフラッディングにより境界線の存在を認識する際のセンサデータ交換を実現している．一方，本章では，各端末がアプリケーション要求に関する情報を事前に所持している環境を想定しているため，クエリのフラッディングは不要である．しかし，センサデータ交換のためには有効な方法であるため，比較手法では，2.4.2 項で述べた方法において，クエリからアプリケーション要求に関する情報を除いたメッセージ（存在認識用メッセージ）のフラッディングにより，各端末が境界線の存在を認識するものとした．

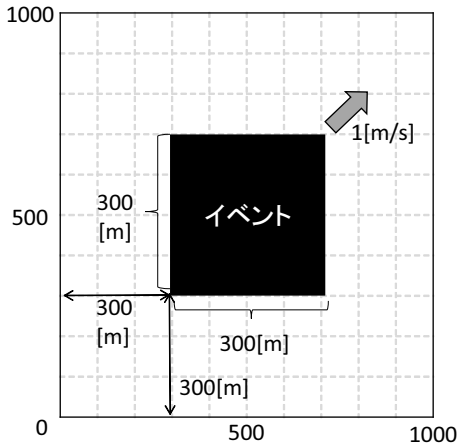


図 3.4: シミュレーション実験で用いるイベント

表 3.1: パラメータ設定

パラメータ (意味)	デフォルト値 (値域)
N (端末数)	1,000 (500∼2,000)
C (観測周期)	30 (15∼480)[s]

3.5.3 パラメータ

本実験で変更したパラメータを表 3.1 に示す．これらのパラメータは，基本的にはデフォルト値に設定されるが，そのパラメータの影響を調べる際には，括弧内の範囲で変化させた．

前項で述べたとおり，提案手法では，各端末が自身の観測値の時間的変化を監視することで，クエリをフラッディングすることなく，境界端末を識別する．この動作によるトラヒックの削減量は，端末数によって変化するものと考えられる．そこで本実験では，端末数 N を変化させて，各手法の性能を調べた．

また，提案手法では，観測時刻間に端末が長距離移動できる場合，境界線から遠く離れた端末が，3.4.2 項で述べた処理によってセンサデータを送信する可能性がある．この影響を調べるため，本実験では，観測周期 C を変化させて，提案手法の性能を調べた．

表 3.2: メッセージサイズ

手法	手順	メッセージ名	サイズ [B]
提案	境界線存在認識 および境界端末識別	データパケット	3 + 24
	データ収集	返信パケット 立候補メッセージ 重複回避メッセージ	24 + 24×（センサデータ数） 18 18
比較	境界線存在認識	存在認識用メッセージ	38
	境界線端末の識別	データパケット	6 + 24
	データ収集	返信パケット	12 + 24×（センサデータ数）
共通	データ収集	ACK	6

3.5.4 評価指標

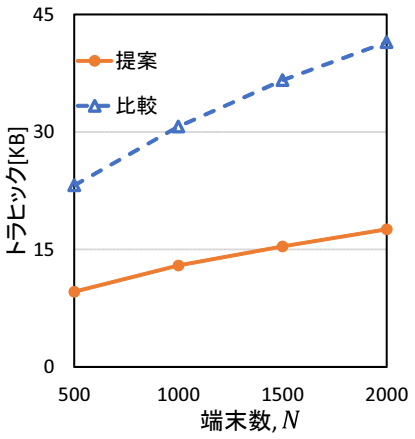
- **トラヒック**：シンクおよびすべての端末が送信したメッセージのアプリケーション層におけるサイズの合計．データ収集 1 回あたりの平均値を測定した．各メッセージサイズを表 3.2 に示す．なお，3.4.2 項の動作による影響を検証するため，3.4.2 項で述べた処理，および比較手法における存在認識用メッセージの伝播に要するトラヒックを，境界線の存在の認識に要したトラヒックとして個別に計測した．

- **推定誤差**：各観測時刻における実際の境界線と，シンクが推定した境界線によって囲まれる領域の面積を，実際の境界線の長さで割った値．また本節でも，2.5 節と同様に，各観測時刻における観測領域内の全端末のセンサ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．

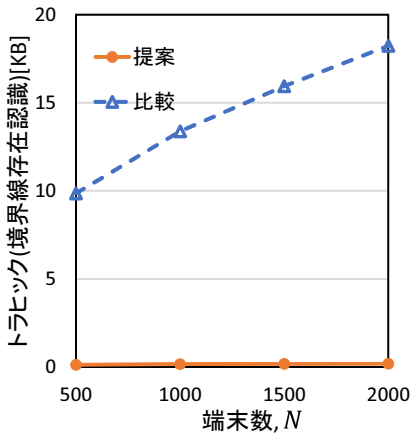
3.5.5 端末数による影響

端末数 N を変化させたときの結果を図 3.5 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 3.5(a) はトラヒック，図 3.5(b) は境界線の存在認識に要したトラヒック，図 3.5(c) は推定誤差を表す．

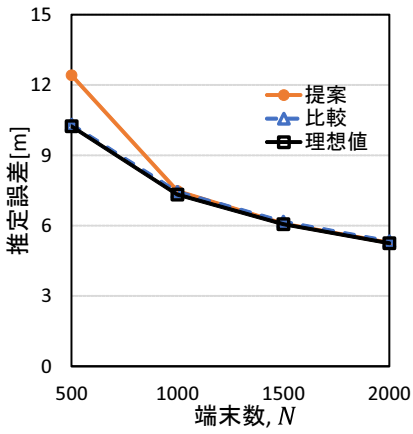
図 3.5(a) より，提案手法のトラヒックが，比較手法より小さいことがわかる．これは，提案手法では，各端末が観測値の時間的変化を監視することで，存在認識



(a) トラヒック



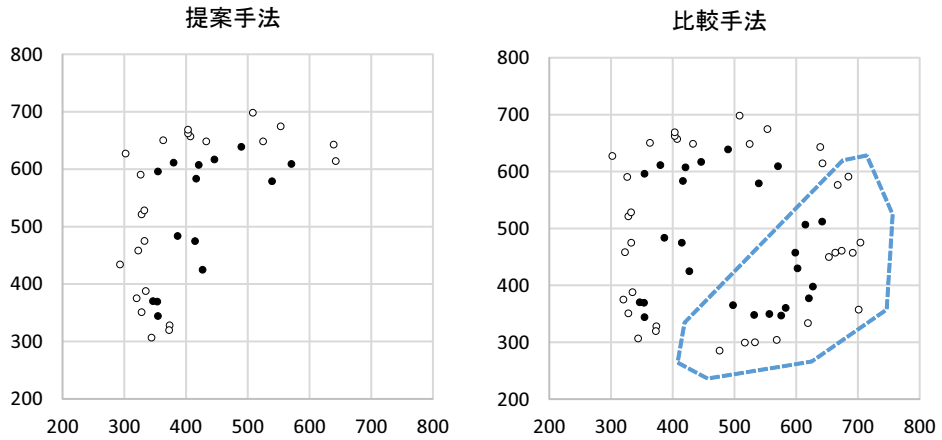
(b) トラヒック（境界線存在認識）



(c) 推定誤差

図 3.5: 端末数による影響

用メッセージのフラッディングなしで，境界端末を識別するためである．さらに，端末数の増加に伴い，トラヒックの削減量が増加することがわかる．これは，比較手法では，境界端末数に加え，存在認識用メッセージを転送する端末も増加するためである．また図 3.5(b) より，前述のとおり，提案手法が，存在認識用メッセージをフラッディングすることなく，観測値の時間的変化を監視を用いることで，境界線の存在把握に要するトラヒックを大幅に削減し，その削減量が端末数の増加に伴って増加していることがわかる．



(a) 提案手法　　(b) 比較手法

図 3.6: 境界端末識別時にセンサデータを送信した端末の分布

図 3.5(c) より，提案手法における推定誤差が，比較手法や理想値と同等の値を示しつつ，推定誤差が端末数の増加に伴って低下していることがわかる．これは，2.5.5 項で述べた理由と同じである．一方，端末数が 500 台のときに，提案手法における推定誤差が，比較手法より大きくなっていることがわかる．これは，3.4.5 項で述べたとおり，端末数が少なくなることで，境界線を跨ぐ端末の数が減少し，一部の境界端末が識別できない場合があるためである．

これを確認するために，端末数が 500 台のときに，各手法において，ある観測時刻における境界端末の識別時にセンサデータを送信した端末の 2 次元平面上における分布を調査した．結果を図 3.6 に示す．これらの結果より，提案手法において，一部の境界端末がセンサデータを送信していないことがわかる．これは，提案手法において，図 3.6(b) における点線で囲まれた範囲周辺のどの端末も，時刻 $(T_i - s)$ から T_i の間に境界線を跨いでおらず，さらに，境界端末の識別中に送信されたセンサデータも受信できなかったためである．しかし，3.4.5 項で述べたとおり，上記はモバイルセンサネットワークを用いてユーザ参加型センシングを行う環境をはじめとする端末が多数存在する環境では発生しにくい問題であり，前述のとおり，端末数が多い場合は，提案手法も比較手法や理想値と同等の推定誤差を示している．

3.5.6 観測周期による影響

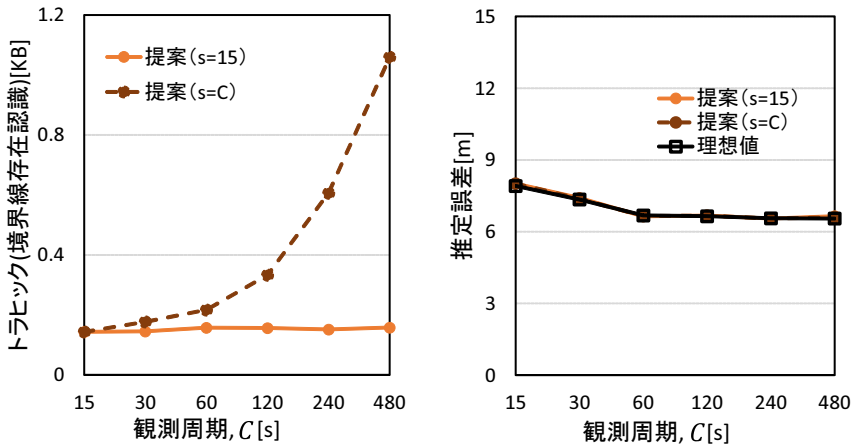
3.5.3 項で述べたとおり，パラメータ s は，各端末が境界線の存在を認識するために要するトラヒックに影響を与える．これを確認するために，提案手法において，$s = 15$（$= c$）に設定したものと，$s = C$ に設定したもののそれぞれの性能を比較した．ここで，$s = C$ に設定した手法では，観測時刻間に境界線を跨いだすべての端末がセンサデータを送信できるため，観測周期 C が長くなった場合，境界線から遠い端末がセンサデータを送信するものと考えられる．

また，本項の実験では，境界線に対する各端末の相対的な速度と s との関係性にのみ焦点を当てて調べるために，イベントの移動速度は 0[m/s] に設定した．

観測周期 C を変化させたときの結果を図 3.7 に示す．グラフの横軸は観測周期 C を表し，縦軸はそれぞれ，図 3.7(a) は境界線の存在認識に要したトラヒック，図 3.7(b) は推定誤差を示す．

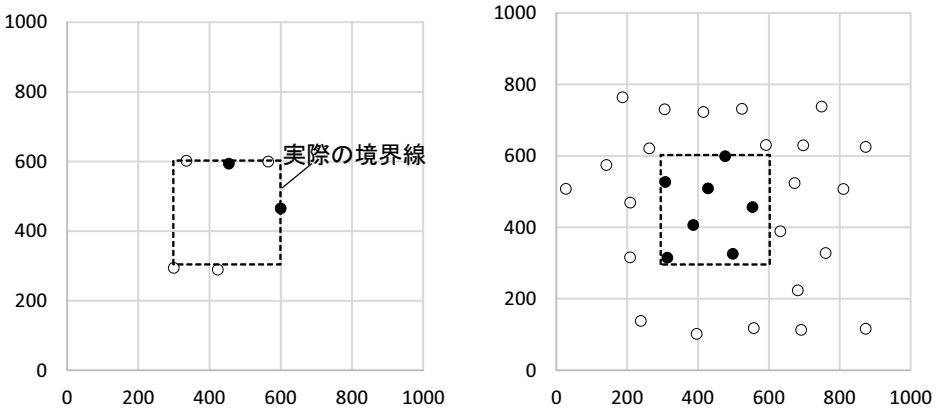
この結果より，観測周期が長くなっても，s を 15[s] に設定した提案手法はトラヒックがほぼ変化しないことがわかる．それに対し，s を観測周期と同じ値に設定した手法は，観測周期が長くなるほど，トラヒックが増加することがわかる．以上から，s を小さく設定することによって，境界線から遠い端末によるセンサデータの送信を抑制できることがわかる．上記を確認するために，$C = 480$[s] のときに，3.4.2 項の処理によってセンサデータを送信した端末の 2 次元平面上における分布を調査した．結果を図 3.8 に示す．図 3.8(a) より，前述のとおり，s を小さく設定することによって，境界線から遠い端末によるセンサデータの送信を抑制できることがわかる．一方，図 3.8(b) より，s を観測周期と同じ値に設定した手法は，境界線から遠い端末もセンサデータを送信していることがわかる．これは，観測時刻 T_{i-1} 終了後すぐに境界線を跨いだ端末が，境界線から離れる方向へ移動してしまったためである．

図 3.7(b) より，各手法における推定誤差が理想値と同等の値を示していることがわかる．この結果より，観測周期が長くなった場合においても，境界端末を定期的に識別できることがわかる．



(a) トラヒック（境界線存在認識）　　(b) 推定誤差

図 3.7: 観測周期による影響



(a) 提案手法（$s = 15$）　　(b) 提案手法（$s = C = 480$）

図 3.8: 境界線の存在認識時にセンサデータを送信した端末の分布

3.6　むすび

本章では，モバイルセンサネットワークにおける観測値の時間的変化の監視を用いた定期的な境界線検出手法を提案した．提案手法では，各端末が定期的に自身の位置における観測値を確認することで，自身の観測値の分類が時間的に変化しているか否か，すなわち自身が境界線を跨いだか否かを監視する．境界線を跨

いだ端末のうち，跨いでからの経過時間が短い端末のみがセンサデータを送信し，そのセンサデータを受信した端末から順に，境界端末の識別処理を行う．これにより，クエリを定期的にフラッディングすることなく，境界端末を識別できる．また，境界端末の中から代表端末を選出し，この代表端末のみが識別時に受信したセンサデータをまとめて，ジオルーティングを用いてシンクへ送信する．ここでセンサデータを送信する前に，代表端末が送信するセンサデータの重複を防ぐため，代表端末間でメッセージ交換を行い，重複しているセンサデータを削除する．これにより，転送木を構築することなく，少ないメッセージ数で境界端末のセンサデータを収集できる．

シミュレーション実験により，提案手法は，第 2 章で提案した手法と比較して，推定誤差を維持しつつ，トラヒックを削減できることを確認した．さらに，端末数が増加するほど，トラヒックの削減量が増加することを確認した．これにより，定期的にイベントの位置と形状を要求するアプリケーションがモバイルセンサネットワークを使用する場合においても，端末のバッテリ消費の低減や，他のアプリケーションのための無線通信帯域の確保が可能となる．

一方，提案手法は，3.4.5 項で述べたとおり，時刻 $(T_i - s)$ から T_i の間に境界線を跨いだ端末が存在しない状況が発生し得る．このとき，本章の提案手法では，境界端末を識別できない．そのため，このような場合においても境界端末を識別できるように，手法の拡張を行う必要がある．また提案手法では，センシング周期 c や，観測時刻付近で境界線を跨いだか否かを判断するためのパラメータ s を予め決定する必要がある．c や s の設定によっては，3.4.5 項で述べたように，境界端末を識別できなくなったり，3.5.6 項で述べたとおり，無駄にトラヒックが増大してしまったりする可能性がある．そこで，これらのパラメータを，境界線の速度や端末の平均的な速度に応じて自動的に設定できるように，手法の拡張を行う予定である．

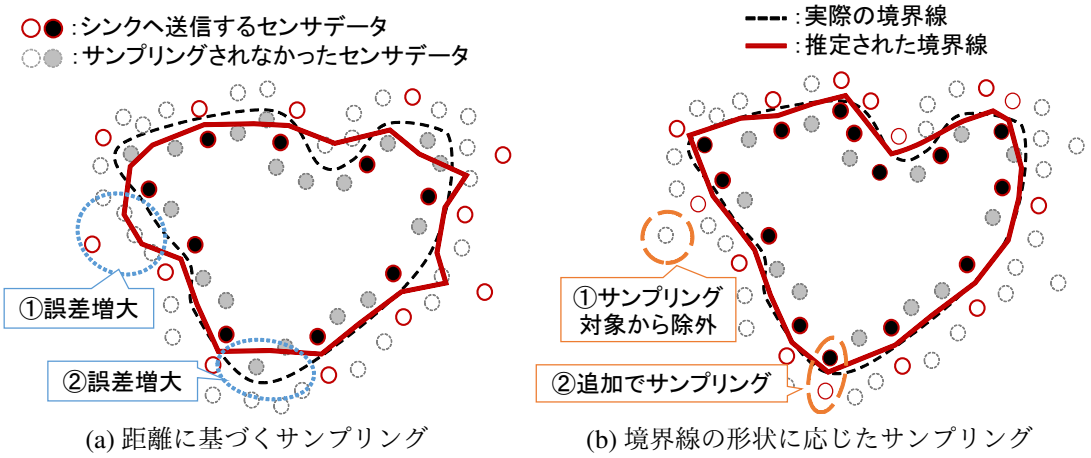
第4章　境界線の形状を考慮したサンプリングを用いた制限時間付き境界線検出手法

4.1　まえがき

第2章で提案した手法は，2.4.5項で述べたとおり，自身が境界端末であると判断したすべての端末からセンサデータを収集するため，境界端末数が極端に多くなる環境では，境界端末のみからセンサデータを収集するとはいえ，大量のトラヒックが発生してしまう．

数百ミリ秒程度のごく短い制限時間内にイベントの位置と形状を要求するアプリケーションに対して，この手法を用いて，与えられた制限時間内に境界線を検出しようとすると，境界端末が非常に多い場合，パケット衝突の頻発によって多くのセンサデータの収集に失敗し，推定する境界線の誤差が増大してしまう．また，結果的に収集に失敗するセンサデータのためのトラヒックが大量に発生し，無線通信帯域を浪費してしまう．通信帯域が有限であるモバイルセンサネットワークでは，このような無駄なトラヒックは削減するべきである．

この問題を解決するためには，制限時間内に収集できる最大センサデータ数を超えないように，データ収集を開始する前にセンサデータ数を調整すればよい．しかし，1.3.3項で述べたとおり，事前に境界端末の数がわからない環境ではこの調整は不可能であるため，端末の自律的な動作により，境界端末を識別しながら，それらの端末が送信するセンサデータの数を可能な限り削減する必要がある．さらに，収集するセンサデータの削減による境界線の推定誤差の増大を可能な限り抑制する必要もある．これらを実現するために，境界端末を地理的にサンプリング



(a) 距離に基づくサンプリング　(b) 境界線の形状に応じたサンプリング

図 4.1: 提案手法によるサンプリング

する手法を提案する．提案手法ではまず，2.4.3 項で説明した処理によって識別される端末を，図 4.1(a) のように，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センサデータをシンクへ送信する．しかし，これらのセンサデータだけで境界線を推定すると，図 4.1(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，このサンプリングによる境界線の推定誤差の増大を抑えるため，送信されたセンサデータを受信した他の端末が，受信したセンサデータを用いて，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，推定誤差の低減が見込める場合にのみ，図 4.1(b) のように，シンクへ収集するセンサデータの追加や除外を行う．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，推定する境界線の誤差の増大を抑制しつつ，収集に失敗するセンサデータのためのトラヒックを削減できる．

以下では，まず 4.2 節で想定環境について述べ，4.3 節で関連研究について述べる．その後，4.4 節で提案手法について述べた後，4.5 節でシミュレーション実験の結果を示す．最後に 4.6 節で本章のまとめと課題について述べる．

4.2 想定環境

端末構成および境界線推定方法は，基本的には2.2節で述べたものと同じであるが，これに加え，アプリケーションは制限時間 t_{end} を設定し，この時間内の境界線検出を要求する．

4.3 関連研究

本節では，固定センサネットワークにおける制限時間付きデータ収集に関する既存研究，および境界線付近の端末のサンプリングを行っている既存研究について述べる．

4.3.1 固定センサネットワークにおける制限時間付きデータ収集

文献 [2, 3, 28] では，制限時間内のデータ収集が要求される環境において，可能な限り多くのセンサデータを収集するための手法を提案している．文献 [3, 28] では，シンクを根，各端末を内部ノードまたは葉ノードとする木構造が既に構築されており，かつ各端末のセンサデータの送信タイミングがスロット単位でスケジュールされる環境を想定している．これらの研究では，上記の環境において，与えられた制限時間内に最も多くセンサデータを収集できる送信スケジュールを作成する手法を提案している．ここで，センサデータの送信タイミングがスロット単位でスケジュールされる環境では，同じ宛先のセンサデータが同じスロットで複数送信されると，パケット衝突が発生し，宛先の端末がそのセンサデータを受信できなくなる．この環境において，これらの手法では，葉ノードから順にセンサデータを集約しながら収集する際に，木構造上で同じ親をもつ（兄弟関係にある）端末が，異なるスロットでセンサデータを送信し，かつセンサデータが送信されるスロットの数が制限時間を超えないように，送信スケジュールを作成する．このとき，送信タイミングの割当て先が，兄弟関係にある端末と同じスロット以外にない端末のセンサデータを，収集対象から除外することで，構築済みの木構造を用いて制限時間内に可能な限り多くのセンサデータを収集する．文献 [2] で提案さ

れている手法では，上記の手法が作成した送信スケジュールによって収集できるセンサデータ数が，構築済みの木構造に依存する点に着目し，与えられた制限時間と端末分布に応じて，より多くのセンサデータを収集できるように，木構造上の親子関係を変更する．

上記の研究は，制限時間付きのデータ収集を考慮している点が本研究と類似するが，境界線検出を対象としていない点が異なる．

4.3.2　固定センサネットワークにおける境界線付近の端末のサンプリング

文献 [38, 39, 66] では，境界線をモニタリングする際のトラヒックを削減するために，境界線付近の端末をサンプリングする手法を提案している．文献 [39, 66] で提案されている手法では，3.4.4 項で述べた方法のように，メッセージの送受信を用いて，境界線付近の端末の中から，互いに通信範囲内に存在しない端末群をサンプリングする．文献 [38] で提案されている手法では，一定の間隔でサンプリングした境界線付近の端末を頂点とする閉路グラフが構築されている環境において，境界線の変化によって，境界線付近の端末が変化した場合，サンプリングした端末間の距離が一定になるように，追加のサンプリングおよびサンプリングした端末の除外を行う．

上記の研究は，境界線付近の端末をサンプリングする点が本研究と類似する．しかし，本研究と異なり，境界線の形状を考慮したサンプリングを行っていないため，境界線の形状によっては，推定誤差が増大してしまう．

4.4　提案手法

本節では，まず提案手法の概要について述べ，その後，提案手法の詳細な手順について説明する．

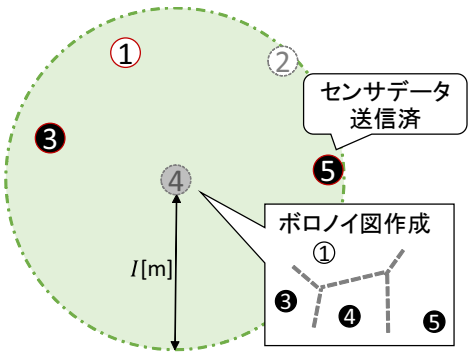


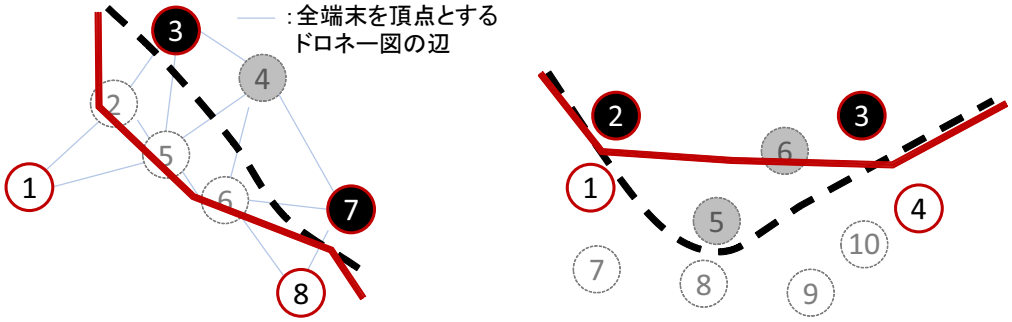
図 4.2: 距離に基づくサンプリングの例

4.4.1 動作概要

提案手法では，第 2 章で提案した手法と同様に，2.4.2 項で述べる方法で，クエリを観測領域全体に伝播する．ここで，クエリには，2.4.2 項で述べた情報に加え，制限時間 t_{end} を付与する．クエリを受信した端末は，4.4.2 項および 4.4.3 項で述べる方法で，境界線の形状に応じて，境界端末を識別しつつ，収集するセンサデータ数を削減する．その後，制限時間内にセンサデータを収集するために，4.4.4 項で述べる方法で，センサデータを収集する．

4.4.2 距離に基づく地理的なサンプリング

基本的な動作は 2.4.3 項と同様であるが，センサデータを送信する端末間の距離が閾値 I[m] 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，ボロノイ図上で観測値の分類が異なる端末と隣接しても，自身を中心とする半径 I（$\leq r$）[m] の円内に観測値の分類が同じセンサデータを送信済みである端末が存在する場合，センサデータの送信を中止する．このときの動作例を図 4.2 に示す．この図は，端末 $\{1, 3, 5\}$ の順にセンサデータを送信済みである状態を示している．これらのセンサデータを受信した端末 4 は，観測値の分類が異なる端末 1 とボロノイ図上で隣接しているが，自身を中心とする半径 I[m] の円内に位置し，自身と観測値の分類が同じ端末 $\{3, 5\}$ がセンサデータを送信済みであるため，センサデータの送信を中止する．



(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 4.3: 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

以上の手順によって，観測値の分類が同じ端末が閾値 I[m] 以上の間隔でサンプリングされ，送信されるセンサ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 4.3(a) において，第 2 章で提案した手法が識別する端末 $\{2, 5, 6\}$ がセンサデータの送信を中止することによって，端末 1 のように，実際には境界端末ではない端末が，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センサデータを送信した場合，推定される境界線が，センサ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，このようなセンサデータの送信によって，図 4.3(a) の端末 2 など，実際の境界端末のセンサデータ送信を妨げてしまう可能性もある．

2. 図 4.3(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．この端末のセンサ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れた境界線を推定してしまう．

4.4.3 境界線の形状に応じたサンプリング

前項で述べた問題を解決するために，各端末は，自身がセンサデータを送信する，または2.4.2項で述べた識別終了時刻t_{detect}になるまで，前項の処理で送信されたセンサデータから，シンクと同じ手順によって部分的に境界線を推定する．これを基に，提案手法は，サンプリング対象になっている境界端末の除外，および境界端末の追加のサンプリングを行う．以下では，その詳細について説明する．なお，以下で述べる処理は，自身が境界端末であると判断した端末のうち，センサデータを送信していない端末によって行われる．

サンプリング対象からの除外

4.4.2項で述べた問題1を解決するためには，各端末が，自身が実際に境界端末である可能性がある否かを，自身が受信したセンサデータのみから判断しなければならない．ここで，ボロノイ図の特性上，ボロノイ領域の境界はドロネー辺の垂直二等分線であり，図4.4における端末7と8のように，互いに近傍に位置する端末間のドロネー辺は，当該端末を母点とするボロノイ領域の境界と交差しやすくなる．一方，図4.4における端末1と7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，ドロネー辺とボロノイ領域の境界が交差しにくくなる．本研究では，ドロネー辺の中点間を結ぶ線分の集合を境界線として推定するため，ドロネー辺とボロノイ領域の境界が交差しない場合，端末1のように，推測される部分的な境界線が，ボロノイ領域と接さない状態になる．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，境界端末では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センサデータ送信を抑制する．具体的には，4.4.2項の処理により，式(2.2)で定義した$T_{boundary}$を設定している端末は，$T_{boundary}$が切れる前に，受信したセンサデータと自身のセンサデータを用いて，2.2.3項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端末ではない可能性があると判断し，センサデータの送信を中止する．

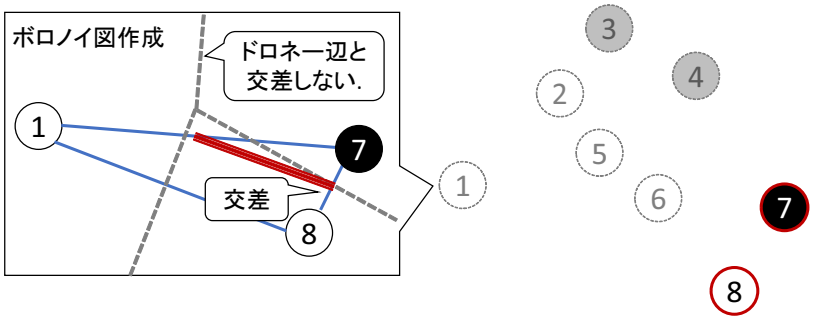


図 4.4: サンプリング対象からの除外の例

追加のサンプリング

4.4.2 項で述べた問題 2 を解決するためには，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のセンサデータを収集すればよい．そこで提案手法では，4.4.2 項の処理によってサンプリング対象にならなかった端末が，受信したセンサデータを用いて，境界線を部分的に推定する．例えば図 4.5 において，端末 $\{1, \ldots, 4\}$ のセンサデータを受信した端末 5 は，これらのセンサ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，自身のセンサデータは境界線の推定に用いない．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，次式により，自身のセンサデータ（修正用データ）を送信するためのタイマ T_{mod} を設定する．

$$T_{mod} = (1 - \frac{d_b}{r}) \cdot T_{mod}^{max}. \tag{4.1}$$

上式の d_b は推定した境界線までの距離である．T_{mod}^{max} はタイマ T_{mod} の最大値であり，全端末共通で予め設定される値である．式 (4.1) より，タイマ T_{mod} は，推定した境界線からの距離が大きいほど短く設定される．このタイマが切れるまでに，自身を中心とする半径 I[m] の円内に位置する端末から，観測値の分類が同じ修正用データを受信しなかった場合にのみ，修正用データであることを示すフラグを付与したセンサデータを，2.4.2 項で述べた転送木上の親へ送信し，本項の全サンプリング処理を終了する．例えば図 4.5 の端末 5 は，部分的に推定した境界線によ

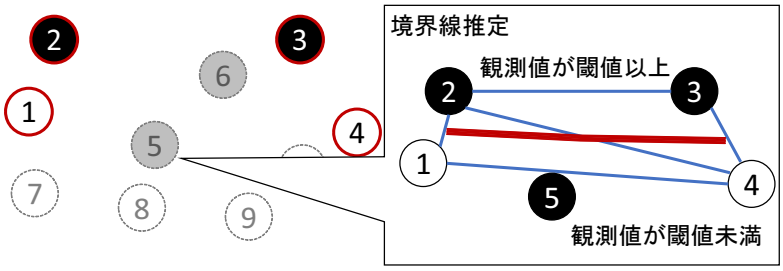


図 4.5: 追加のサンプリングの例

る，自身の位置の観測値の分類が「閾値未満」であり，実際の観測値の分類と異なっているため，この境界線から遠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．

この修正用データを用いて推定される境界線を修正するためには，修正用データを送信した端末とドロネー図上で隣接する，観測値の分類が異なる端末のセンサデータが必要である．そこで，上記の処理で送信された修正用データを受信した端末のうち，送信元端末と観測値の分類が異なる端末は，自身のセンサデータを送信するためのタイマ $T_{pair-mod}$ を，式 (2.2) と同様に，送信元端末に近いほど短くなるように設定する．上記のタイマの設定は，4.4.2 項の処理によってサンプリング対象になっているか否かに関わらず行う．また，既に $T_{boundary}$ を設定している場合，$T_{boundary}$ を止め，代わりに $T_{pair-mod}$ を設定する．$T_{pair-mod}$ が切れるまでに，自身を中心とする半径 I[m] の円内に位置する端末のうち，センサデータを送信していない端末から，観測値の分類が同じセンサデータを新たに受信しなかった場合にのみ，自身のセンサデータを転送木上の親へ送信し，本項のサンプリング処理を終了する．例えば図 4.5 において，端末 5 が送信した修正用データを受信した端末 8 は，端末 5 に近いほど短くなるタイマを設定し，自身のセンサデータを送信する．これにより，図 4.6 に示すように，端末 5 の位置の観測値を正しく分類できる境界線を推定できる．

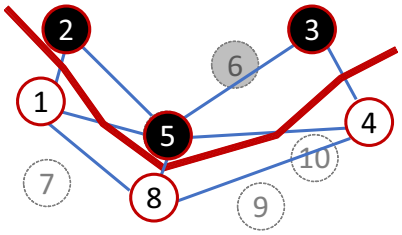


図 4.6: 端末 $\{5, 8\}$ のセンサデータにより修正された境界線

4.4.4　データ収集

第 2 章で提案した手法では，制限時間が指定されることを想定しておらず，境界端末であると判断した端末のセンサデータを，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．この方法で制限時間内のデータ収集を実現しようとした場合，収集用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，短期間にメッセージの送信が集中し，パケット衝突および再送が頻発する．結果として，シンクが複数のメッセージを制限時間内に受信できなくなる．特に第 2 章で提案した手法では，2.4.4 項で述べた処理によって，複数のセンサデータを一つのメッセージに集約して収集するため，メッセージの収集失敗によって多数のセンサデータが推定に使用できなくなる．また，1 メッセージあたりのサイズが大きくなるため，メッセージ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センサ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．

そこで提案手法では，シンクが可能な限り多くのセンサデータを受信できるようにするために，個々のセンサデータを集約せずに，2.4.2 項で述べた転送木に従って収集する．ここで，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センサデータは，識別終了時刻 t_{detect} に送信される．このセンサデータを制限時間 t_{end} 内にシンクが受信するためには，　このセンサデータの収集を $(t_{end} - t_{detect})$ の期間内に行う必要がある．これを実現するため，以下の式に従っ

て，受信したセンサ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．

$$W_g' = \frac{t_{end} - t_{detect}}{h_{max}} - delay_{trans} + \mathit{rand}. \tag{4.2}$$

ここで，h_{max} はクエリに付与されていたシンクからのホップ数の最大値，$delay_{trans}$ は 1 ホップあたりの転送遅延を示す．また，*rand* はパケット衝突を回避するために与える乱数である．上式の第 1 項は，$(t_{end} - t_{detect})$ の期間内に 1 ホップあたりのセンサ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，自身がセンサデータを受信してから，そのセンサ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．転送木上の内部ノードに該当する端末は，サンプリングされたセンサデータを受信するたびに，受信してから W_g' 経過後，受信したセンサデータを自身の転送木上の親へ送信する．また，再送の頻発によるトラヒックの増大を防ぐため，センサデータの再送は行わないものとする．上記の処理により，識別終了時刻 t_{detect} までに送信されたセンサデータを，パケット衝突が発生しない限り，シンクは制限時間内に受信できる．

4.4.5 考察

提案手法では，まず境界端末を等間隔にサンプリングし，サンプリングされた端末のみがセンサデータを送信するように制御する．また，送信されたセンサデータを受信した他の端末は，受信したセンサデータから部分的に推定した境界線を基に，シンクへ送信するセンサデータの追加や除外を行う．この処理によって，提案手法は，推定する境界線の誤差の増大を抑制しつつ，収集に失敗するセンサデータのためのトラヒックの削減を図っている．しかし，これを実現できるか否かは，4.4.2 項で述べた閾値 I の大きさに依存する．例えば，I を小さく設定すると，多くの端末が 4.4.2 項の処理によってサンプリング対象になるため，4.4.3 項の処理を行う端末数が減少する．また，シンクへ送信するセンサデータ数も増加するため，パケット衝突が発生しやすくなり，制限時間内では収集できないセンサデータ数が増加する．一方，I を大きく設定すると，シンクへ送信するセンサデータ数が減少するため，パケット衝突が起こりにくくなり，制限時間内では収集できないセン

サデータ数が減少する．しかし，各端末において自身の通信範囲内でセンサデータを送信する端末も減少するため，自身のセンサデータなしでは境界線を推定できない端末が発生し，4.4.3 項で述べた追加のサンプリングができなくなり，4.4.2 項で述べた問題が発生してしまう．このように閾値 I は，境界線の推定誤差と，制限時間内では収集できないセンサデータ数に影響を与えるものと考えられる．

また，4.4.3 項で述べた追加のサンプリングがどれくらい行われるかは，I の設定に加え，端末密度にも依存するものと考えられる．例えば端末数が非常に多い場合，部分的に推定した境界線を基に，自身の位置における観測値の分類が正しいかどうかを判断する端末が増えるため，追加でセンサデータを送信する端末数が増加するものと考えられる．

これらの影響については，4.5 節において詳細に検証する．

4.5　性能評価

本節では，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，変化が速いイベントの境界線のリアルタイムな検出が求められる環境における，提案手法の性能を評価するために，PM2.5 や花粉などの微粒子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短い時間で検出し，ネットワークに参加しているユーザに対してマスク着用の注意喚起をする状況を想定した．実験では，2.5 節と同じネットワークシミュレータを用いた．

4.5.1　シミュレーション環境

観測領域や端末構成は，基本的には 2.5.1 項で述べたものと同じであるが，シンクは，シミュレーション開始から制限時間 t_{end} 内にデータ収集を行うものとする．また，識別終了時刻 t_{detect} は，制限時間未満になるように設定した．式 (4.1) を用いてタイマを算出する際の定数 T_{mod}^{max} は，識別終了時刻 t_{detect} が 0.3[s] のときに，識別終了時刻までに境界端末の識別およびサンプリング処理が終了するように，0.05[s]

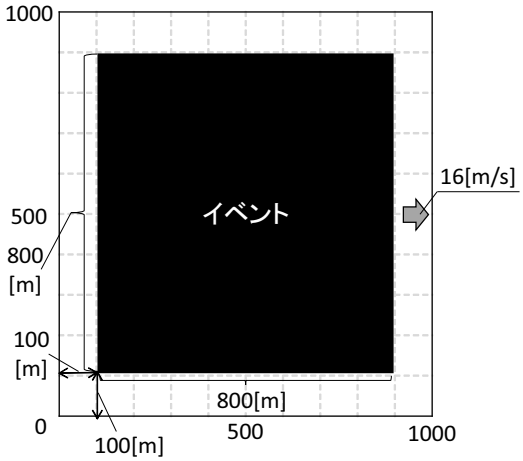


図 4.7: シミュレーション実験で用いるイベント

とした．また，式 (4.2) において，パケット衝突を回避するための乱数 *rand* は，[0, 10^{-5}][s] の範囲内でランダムに決定するものとした．

図 4.7 に，本実験で使用したイベントの形状を示す．このイベントは，一辺が 800[m] の正方形であり，図の位置を初期位置とする．また，PM2.5 や花粉などの微粒子の濃度が高い領域が風に乗って移動する [69] ときの境界線の速度を想定して，矢印の方向に 16[m/s] の速度で等速移動するものとした．

4.5.2 評価指標

- **総トラヒック**：シンクがクエリを発行した時刻から t_{end} 内に，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合計．各メッセージのサイズを表 4.1 に示す．

- **無駄なトラヒック**：シンクが受信できなかったセンサデータの送信に要した総トラヒック．

- **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の面積を，実際の境界線の長さで割った値．また本節でも，2.5 節および 3.5 節と同様に，クエリ発行時刻における観測領域内の全端末のセンサデータを

表 4.1: 提案手法におけるメッセージサイズ

手順	メッセージ名	サイズ [B]
クエリ伝播	クエリ	92
境界端末識別およびサンプリング	データパケット	21 + 24
データ収集	返信パケット	26 + 24

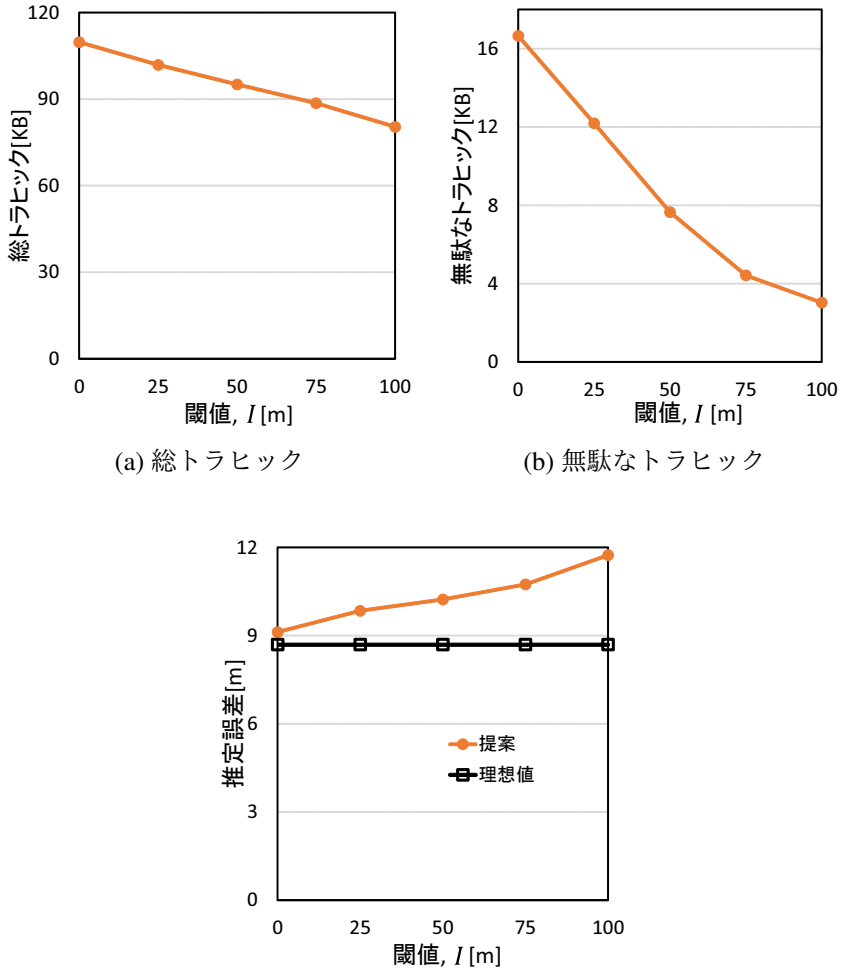
用いて推定した境界線の推定誤差を，理想値として個別に計測した．

4.5.3　提案手法の特性

4.4.5 項で述べたとおり，提案手法の性能は，4.4.2 項で述べたサンプリング時の閾値 I による影響を受けるものと考えられる．そこで，I を [0, 100][m] の範囲で変化させ，4.5.2 項で述べた各評価値を計測した．制限時間 0.4[s]，識別終了時刻 0.3[s]，端末数 1,000 台のときの結果を，図 4.8 に示す．グラフの横軸は閾値 I を表し，縦軸はそれぞれ，図 4.8(a) は総トラヒック，図 4.8(b) は無駄なトラヒック，図 4.8(c) は推定誤差を表す．

図 4.8(a) より，I が大きくなるほど，トラヒックが減少していることがわかる．これは，I が大きくなるほど，シンクへ送信するセンサデータ数が減少するためである．図 4.8(b) より，I が大きくなるほど，無駄なトラヒックが減少していることがわかる．これは，上記で述べたとおり，I が大きくなるほど，シンクへ送信されるセンサデータ数が減少し，それに伴ってパケット衝突が発生しにくくなるためである．

図 4.8(c) より，I が大きくなるほど，推定誤差が増大していることがわかる．これは，I が大きくなるほど，シンクへ送信されるセンサデータ数が減少し，推定時に使用できるセンサデータ数が減るためである．また，4.4.5 項で述べたとおり，4.4.3 項で述べた追加のサンプリングが行われにくくなり，境界線の修正ができなくなるためである．これを確認するために，4.4.2 項で述べた処理によって送信されたセンサデータ数と，4.4.3 項で述べた処理によって送信された修正用データ数を個別に計測した．結果を図 4.9 に示す．図 4.9(a) より，I が大きくなるほど，送信されるセンサデータ数が大幅に減少することがわかる．一方，図 4.9(b) より，I



(a) 総トラヒック

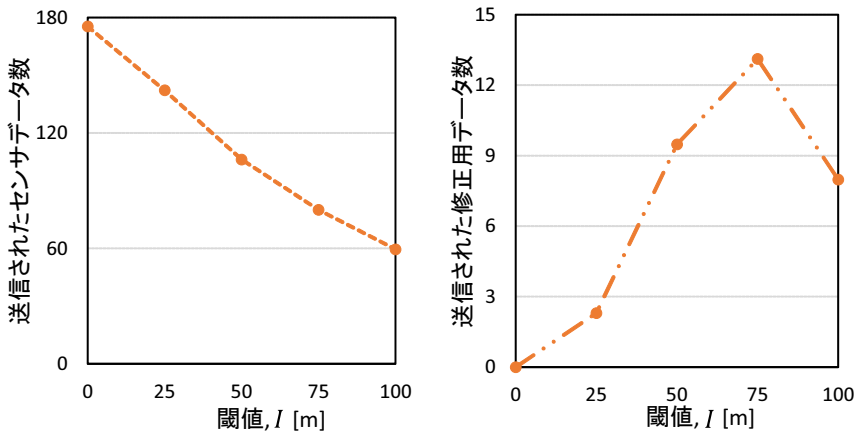
(b) 無駄なトラヒック

(c) 推定誤差

図 4.8: 制限時間による影響

が 75[m] になるまでは，I が大きくなるほど送信される修正用データ数が増加するが，それ以降は，I が大きくなるほど，送信される修正用データ数が減少していることがわかる．この結果より，I を大きく設定すると，自身のセンサデータなしでは境界線を推定できない端末が増加し，追加のサンプリングが行われにくくなることがわかる．その結果，4.4.2 項で述べた問題が発生しやすくなり，推定誤差が増大する．

以上の結果より，提案手法において，総トラヒックと推定誤差はトレードオフの関係にあり，閾値 I はこれを制御できるパラメータであることを確認した．



(a) 送信されたセンサデータ数　　(b) 送信された修正用データ数

図 4.9: 境界端末の識別およびサンプリング時に送信されたセンサデータ数，および修正用データ数

4.5.4　提案手法の有効性

パラメータ

本実験で変更したパラメータを表 4.2 に示す．これらのパラメータは，基本的にはデフォルト値に設定されるが，そのパラメータの影響を調べる際には，括弧内の範囲で変化させた．

提案手法では，端末間の距離に基づくセンサデータのサンプリングと，各端末が部分的に推定した境界線に基づくサンプリング対象のセンサデータの追加や除外により，制限時間内に収集しきれないセンサデータ数を可能な限り削減することで，無駄なトラヒックを削減する．一方，制限時間の長さによって，データ収集に使用できる時間が変化するため，制限時間内にシンクが収集できるセンサデータ数が変化するものと考えられる．つまり，提案手法の動作による無駄なトラヒックの削減量は，制限時間の長さによって変化するものと考えられる．そこで本実験では，制限時間 t_{end}[s] を変化させて，各手法の性能を調べた．また識別終了時刻 t_{detect} は，制限時間 t_{end} が長くなるほど，データ収集に使用できる時間 $(t_{end} - t_{detect})$ が長くなるように，表 4.3 に従って設定した．

また，制限時間内に収集しきれないセンサデータ数は，境界端末の数によって

表 4.2: パラメータ設定

パラメータ (意味)	デフォルト値 (値域)
t_{end} (制限時間)	0.4 (0.2∼1)[s]
N (端末数)	1,000 (500∼3,000)

表 4.3: 制限時間および識別終了時刻

制限時間 t_{end}[s]	0.2	0.4	0.6	0.8	1
識別終了時刻 t_{detect}[s]	0.175	0.3	0.3	0.3	0.3
データ収集に使用できる時間 $(t_{end} - t_{detect})$[s]	0.025	0.1	0.3	0.5	0.7

も変化する．そこで本実験では，端末数 N を変化させて，各手法の性能を調べた．

評価手法

本項では，提案手法における有効性を確認するために，評価手法として，以下の三つの手法の性能を調べた．比較手法における各メッセージのサイズを表 4.4 に示す．

- **提案手法**: 4.4 節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センサデータを収集する手法．4.4.2 項で述べた閾値 I は，制限時間 0.4[s]，端末数 1,000 台のときの第 2 章の手法とほぼ同等の推定誤差になる設定として，50[m] を用いた．

- **従来手法**: 第 2 章で述べた手法に従い，境界端末であると判断したすべての端末のセンサデータを収集する手法．ただし，制限時間内にデータ収集を完了させるため，データ収集は 4.4.4 項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**: 提案手法において，4.4.3 項で述べた処理を行わない手法．4.4.2 項で述べた閾値 I は，準備実験を行い，制限時間 0.4[s]，端末数 1,000 台のときの提案手法とほぼ同等の総トラヒックになる設定として，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センサ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．

表 4.4: 比較手法におけるメッセージサイズ

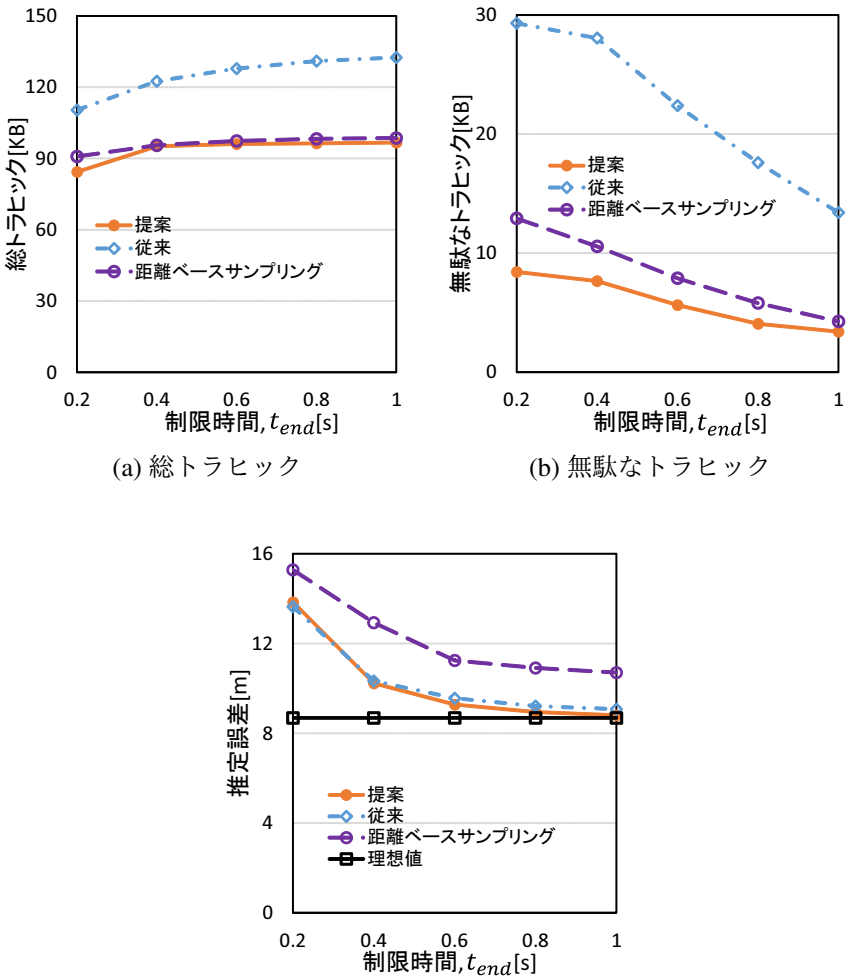
手法	手順	メッセージ名	サイズ [B]
2 章の手法および 距離ベースサンプリング	境界端末の識別 およびサンプリング	データパケット	20 + 24
共通	クエリ伝播 データ収集	クエリ 返信パケット	92 26 + 24

制限時間による影響

制限時間 t_{end} を変化させたときの結果を図 4.10 に示す．グラフの横軸は制限時間 t_{end} を表し，縦軸はそれぞれ，図 4.10(a) は総トラヒック，図 4.10(b) は無駄なトラヒック，図 4.10(c) は推定誤差を示す．

図 4.10(a) より，制限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，制限時間が長くなるほど，パケット衝突の回数が減り，より多くのセンサデータがシンクへ送信されるためである．また，制限時間が 0.2[s] のとき，提案手法のトラヒックが距離ベースサンプリングより小さくなっていることがわかる．これは，一部の修正用データの送信が，識別終了時刻までに間に合わず，シンクへ送信されないためである．さらに，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するセンサデータ数を削減しているためである．これを確認するために，制限時間 0.4[s] における総トラヒックの内訳を調査した．結果を図 4.11 に示す．この結果より，提案手法と距離ベースサンプリングが，従来手法と比較して，境界端末の識別およびサンプリングに要するトラヒックを削減していることがわかる．またこれにより，提案手法と距離ベースサンプリングにおいてデータ収集に要するトラヒックが，従来手法より小さくなっている．

図 4.10(b) より，制限時間が長くなるほど，各手法における無駄なトラヒックが減少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増え，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無駄なトラヒックが，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，シンクへ



(a) 総トラヒック

(b) 無駄なトラヒック

(c) 推定誤差

図 4.10: 制限時間による影響

送信されるセンサデータ数が減り，パケット衝突の回数が減るためである．

図 4.10(c) より，制限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，制限時間が長くなるほど，シンクが多くのセンサデータを受信でき，境界線推定時に使用できるセンサ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プリングによって，境界線の形状を考慮せずにセンサデータを送信する端末がサンプリングされ

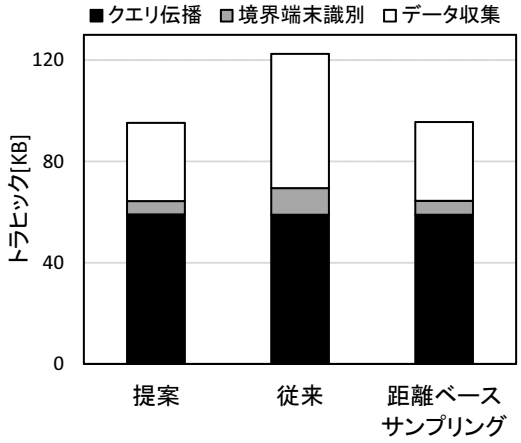


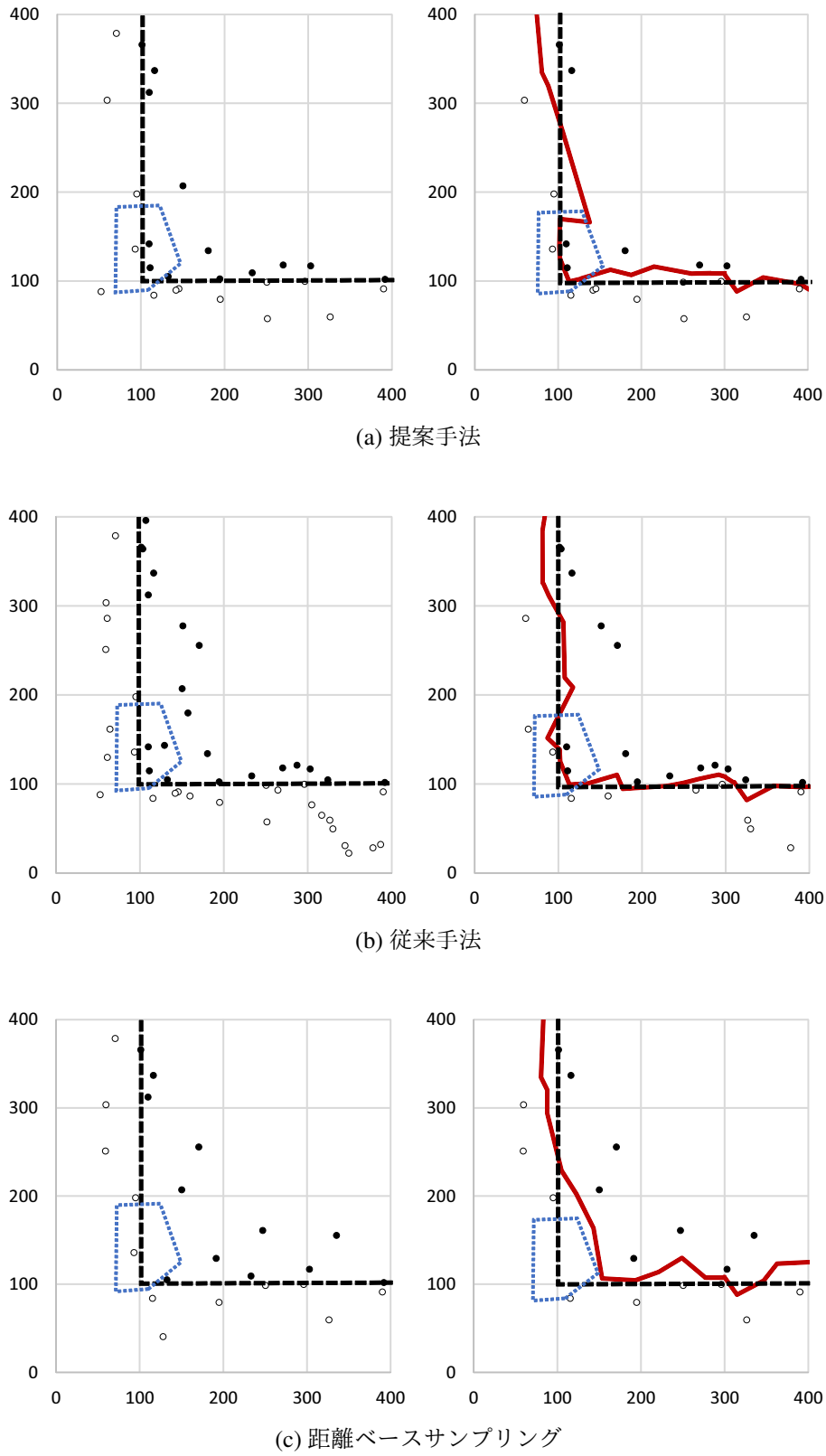
図 4.11: 総トラヒックの内訳

るため，4.4.2 項で述べた問題が発生し，推定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追加および除外することで，単純な距離ベースのサンプリングによる推定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．

推定誤差の差異をより詳細に確認するために，制限時間が 0.4[s] のときに，各手法において，境界端末の識別およびサンプリング時にセンサデータを送信した端末の 2 次元平面上における分布を調査した．さらに，このときシンクが受信したセンサデータの分布，およびそれを基に推定した境界線も調査した．結果を図 4.12 に示す．図 4.12(c) より，距離ベースサンプリングでは，点線で囲まれた範囲内の端末がほとんどサンプリングされていないことがわかる．そのため，推定された境界線が実際の境界線から離れてしまう．一方，図 4.12(a) より，提案手法では，この範囲の端末が複数台サンプリングされていることがわかる．また図 4.12(b) より，従来手法においても，この範囲の端末がセンサデータを送信していることがわかる．そのため，推定された境界線が実際の境界線に近づく．

端末数による影響

端末数 N を変化させたときの結果を図 4.13 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 4.13(a) は総トラヒック，図 4.13(b) は無駄なトラヒッ

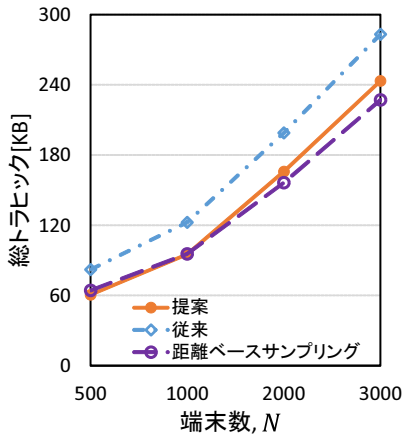


(a) 提案手法

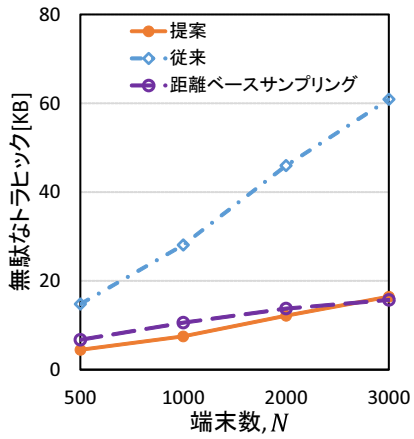
(b) 従来手法

(c) 距離ベースサンプリング

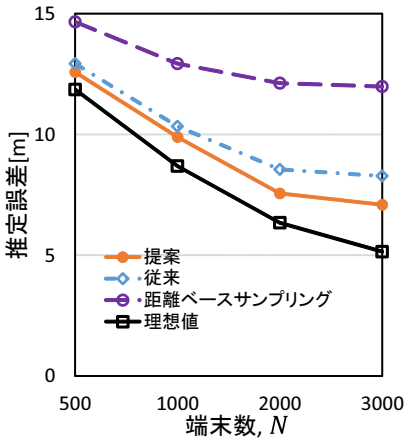
図 4.12: 境界端末の識別およびサンプリング時にセンサデータを送信した端末の分布（左），およびシンクが受信したセンサデータの分布とそれを基に推定した境界線（右）



(a) 総トラヒック



(b) 無駄なトラヒック



(c) 推定誤差

図 4.13: 端末数による影響

ク，図 4.13(c) は推定誤差を示す．

図 4.13(a) より，端末数の増加に伴い，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，境界端末数の増加により，より多くのセンサデータがシンクへ送信されるためである．また，端末数が 1,000 台以上のとき，提案手法におけるトラヒックが，距離ベースサンプリングより大きくなることがわかる．これは，提案手法において，部分的に推定した境界線を基に，自身の位置における観測値の分類が正しいかどうかを判断する端末が増え，シンクへ送信される修正

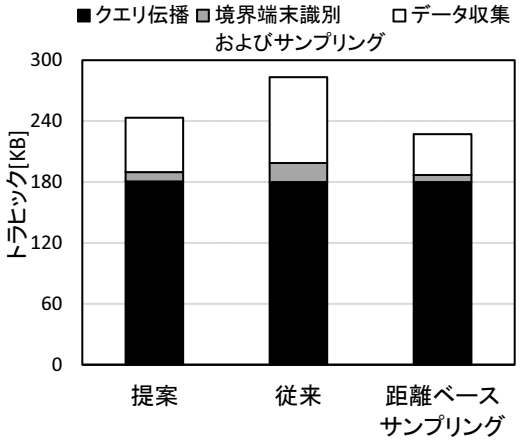


図 4.14: 総トラヒックの内訳

用データが増えるためである．これを確認するために，端末数 3,000 台における総トラヒックの内訳を調査した．結果を図 4.14 に示す．この結果より，提案手法において，クエリ伝播以外の処理に要するトラヒックが，距離ベースサンプリングより大きいことがわかる．これは，提案手法が，距離ベースサンプリングより多くの境界端末をサンプリングし，それらのセンサデータを収集していることを意味する．

図 4.13(b) より，端末数の増加に伴い，各手法における無駄なトラヒックが増加していることがわかる．これは，境界端末数の増加により，より多くのセンサデータがシンクへ送信され，パケット衝突の回数が増えるためである．また，提案手法および距離ベースサンプリングにおける無駄なトラヒックが，従来手法より小さいことがわかる．これは，制限時間を変化させた結果に対して述べた理由と同じである．さらに，端末数の増加に伴い，提案手法と距離ベースサンプリングとの無駄なトラヒックの差が小さくなることがわかる．これは，提案手法において，シンクへ送信されるセンサデータ数が増えるためである．

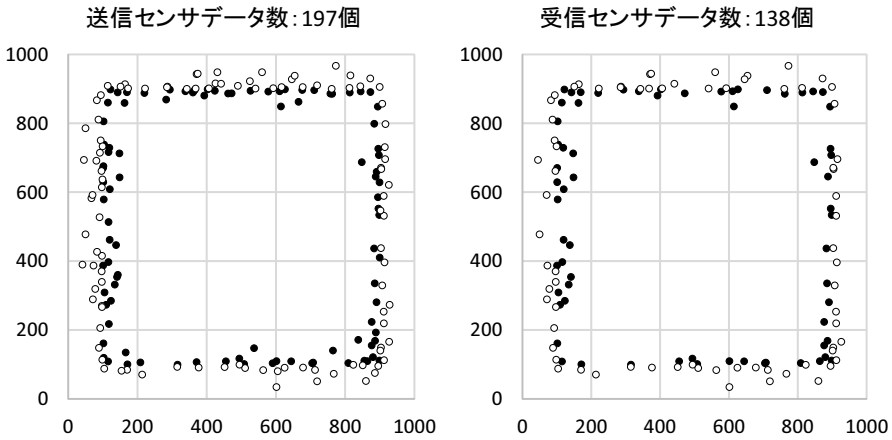
図 4.13(c) より，端末数の増加に伴い，各手法における推定誤差が低下していることがわかる．これは，パケット衝突の回数が増加している一方で，シンクが受信できるセンサデータ数も増加し，かつセンサデータに含まれる位置がより実際の境界線に近づくためである．また，距離ベースサンプリングにおける推定誤差

が，従来手法より大きくなることがわかる．これは，制限時間を変化させた結果に対して述べた理由と同じである．一方，端末数の増加に伴い，提案手法における推定誤差が従来手法より小さくなり，その差が大きくなっていることがわかる．これは，従来手法において，増加した境界端末のセンサデータをすべて収集しようとした結果，通信帯域が圧迫されてしまい，パケット衝突が頻発するためである．その結果，より実際の境界線に近い位置にいる境界端末がセンサデータを送信しているにも関わらず，シンクがそのセンサデータを受信できず，推定精度が小さくなりにくくなる．これに対し提案手法は，4.4.2 項で述べた処理によって収集するセンサデータ数を削減している分，帯域が圧迫されにくいため，より実際の境界線に近い位置にいる境界端末が送信したセンサデータを受信できる．

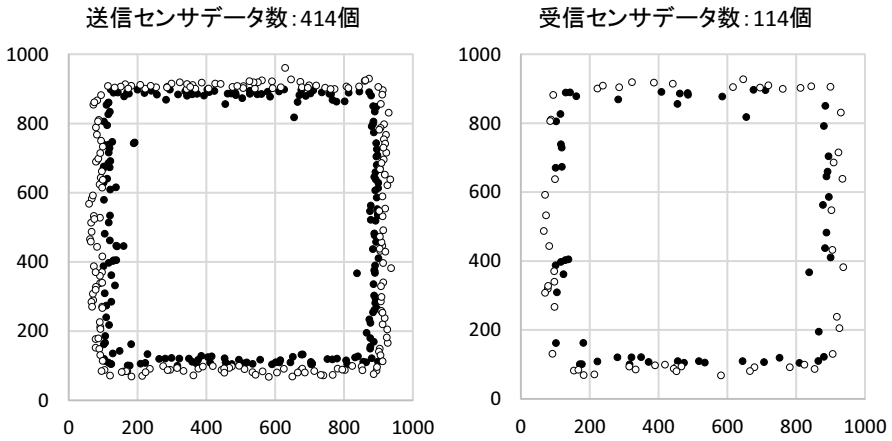
これを確認するために，端末数が 3,000 台のときに，各手法において，境界端末の識別およびサンプリング時にセンサデータを送信した端末の分布，およびシンクが受信したセンサデータの分布を調査した．結果を図 4.15 に示す．図 4.15(b) より，従来手法では，多くの端末が識別時にセンサデータを送信している一方で，その半分以上のセンサデータをシンクが受信できていないことがわかる．それに対し，提案手法と距離ベースサンプリングは，図 4.15(a) および図 4.15(c) より，識別またはサンプリング時に送信されたセンサデータの多くをシンクが受信できていることがわかる．特に提案手法においては，シンクが受信できたセンサデータ数が，従来手法より多いことがわかる．この結果より，提案手法は，4.4.2 項で述べた処理によって収集するセンサデータ数を減らすことで，帯域の圧迫を抑制しつつ，4.4.3 項で述べた処理によって，必要に応じて収集するセンサデータを追加できていることがわかる．

4.6　むすび

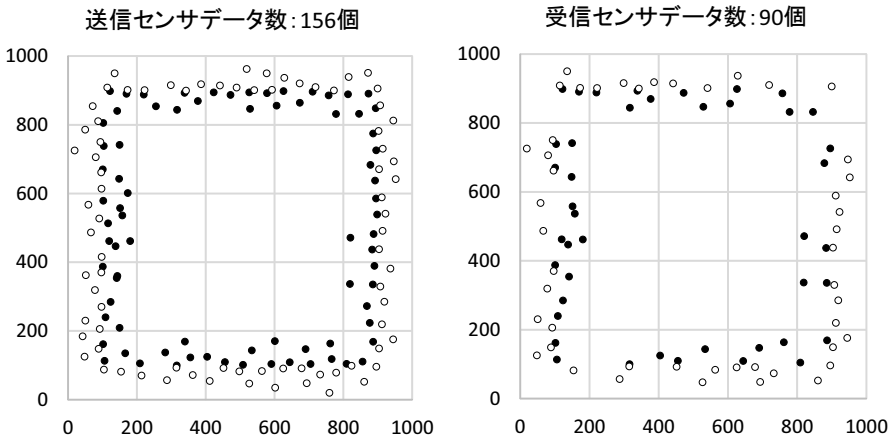
本章では，モバイルセンサネットワークにおける境界線の形状を考慮したサンプリングを用いた制限時間付き境界線検出手法を提案した．提案手法では，制限時間内に収集しきれないセンサデータ数を可能な限り削減するために，境界端末を地理的にサンプリングする．この処理およびサンプリングされた端末のセンサ



(a) 提案手法



(b) 従来手法



(c) 距離ベースサンプリング

図 4.15: 境界端末の識別およびサンプリング時にセンサデータを送信した端末（左），およびシンクが受信したセンサデータ（右）の分布

データの収集を制限時間内に実現するため，提案手法では，制限時間を，境界端末をサンプリングするための時間と，サンプリングした端末のセンサデータをシンクへ送信するための時間との二つに分割する．まず，境界端末をサンプリングするために割り当てられた時間内において，第 2 章で提案した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センサ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誤差の増大を抑制するため，送信されたセンサデータを受信した他の端末が，受信したセンサデータを用いて，自身の周辺における境界線を部分的に推定する．各端末はこれを基に，推定誤差の低減が見込める場合にのみ，シンクへ送信するセンサデータの追加や除外を行う．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，推定される境界線の誤差の増大を抑制しつつ，収集に失敗するセンサデータのためのトラヒックを削減できる．その後，センサデータを収集するために割り当てられた時間内にシンクがセンサデータを受信できるように，センサデータを収集する．

シミュレーション実験により，提案手法は，境界端末であると判断したすべての端末のセンサデータを収集する手法と比較して，推定誤差を維持しつつ，無駄なトラヒックを削減できることを確認した．さらに，境界端末の数が極端に多い環境では，境界端末であると判断したすべての端末のセンサデータを収集する手法より推定誤差が小さくなることを確認した．これにより，モバイルセンサネットワークに参加している端末のバッテリ消費や無線通信帯域への負担を低減しつつ，ネットワークに参加している端末が多い環境において，より精度のよい境界線をユーザに提示できる．

一方，提案手法では，端末間の距離に基づいたサンプリングを行うためのパラメータ I を予め決定する必要がある．しかし，4.4.5 項での議論や，4.5.3 項で示した結果より，パラメータ I は提案手法の性能に大きな影響を与える．そこで，パラメータ I を環境に応じて自動的に設定できるように，手法の拡張を行う必要がある．

第5章　結論

5.1　本論文のまとめ

本論文では，モバイルセンサネットワークにおける境界線検出手法について議論した．

まず，第1章では，近年注目されているユーザ参加型センシングと，それを実現する方法の一つであるモバイルセンサネットワークの特徴について述べた．また，モバイルセンサネットワークにおける境界線の推定を効率的に実現するために，境界端末を識別し，それらの端末のみからセンサデータを収集する境界線検出が重要であることを明らかにし，これを行う際の問題点について述べた．

第2章では，ある時刻におけるイベントの位置と形状を要求するアプリケーションに対して，ドロネー図の双対関係にあるボロノイ図の逐次作成を用いて，事前にドロネー図を構築することなく境界線を検出する手法を提案した．提案手法ではまず，一部の端末のみのメッセージ転送によって，すべての端末へメッセージを伝播できる方法を用いて，アプリケーション要求を含むクエリを，すべての端末へ転送する．また提案手法では，各端末がクエリを送信（あるいは中継）する際，自身のセンサデータをクエリに付与する．これにより，クエリを受信した端末が，自身の通信範囲内に境界線が存在するかどうかを認識できる．境界線の存在を認識した端末から順に，受信したセンサデータを基に，各端末が部分的にボロノイ図を逐次作成し，自身が観測値の分類が異なる端末とボロノイ図上で隣接するか否かを確認する．その結果，境界端末であると判断した場合にのみ，自身のセンサデータを送信する．また，センサデータを送信するための待ち時間を，観測値の分類が異なる端末との距離に基づいて調整することで，最終的に境界端末となる端末のみがセンサデータを送信するように制御する．最後に提案手法では，識

別した境界端末のセンサデータをシンクへ送信する．さらに，提案手法の有効性を示すために，シミュレーション実験による性能評価を行った．その結果より，提案手法は，事前にドロネー図を構築する手法と比較して，推定誤差を維持しつつ，トラヒックを削減できることを確認した．さらに，端末数が増加するほど，トラヒックの削減量が増加することを確認した．

第 3 章では，イベントの位置と形状を定期的に要求するアプリケーションに対して，各端末が自身の観測値の時間的変化を監視することで，クエリを定期的にフラッディングすることなく，境界線を検出する手法を提案した．提案手法では，各端末が定期的に自身の位置における観測値を確認することで，自身の観測値の分類が変化しているか否か，すなわち自身が境界線を跨いだか否かを監視する．境界線を跨いだ端末のうち，跨いでからの経過時間が短い端末のみがセンサデータを送信し，そのセンサデータを受信した端末から順に，境界端末の識別処理を行う．また，境界端末の中から代表端末を選出し，この代表端末のみが識別時に受信したセンサデータをまとめて，ジオルーティングを用いてシンクへ送信する．ここでセンサデータを送信する前に，代表端末が送信するセンサデータの重複を防ぐため，代表端末間でメッセージ交換を行い，重複しているセンサデータを削除する．これにより，転送木を構築することなく，少ないメッセージ数で境界端末のセンサデータを収集できる．さらに，提案手法の有効性を示すために，シミュレーション実験による性能評価を行った．その結果より，提案手法は，第 2 章で提案した手法と比較して，推定誤差を維持しつつ，トラヒックを削減できることを確認した．さらに，端末数が増加するほど，トラヒックの削減量が増加することを確認した．

第 4 章では，制限時間内にイベントの位置と形状を要求するアプリケーションに対して，制限時間内に収集しきれないセンサデータ数を可能な限り減らすために，境界端末を地理的にサンプリングする手法を提案した．提案手法では，制限時間を，境界端末をサンプリングするための時間と，サンプリングした端末のセンサデータをシンクへ送信するための時間との二つに分割し，境界端末をサンプリングするために割り当てられた時間内において，境界端末を等間隔にサンプリングし，その端末のみがセンサデータを送信するように制御する．また，送信さ

れたセンサデータを受信した他の端末は，受信したセンサデータを用いて，自身の周辺における境界線を部分的に推定する．各端末はこれを基に，境界線の推定誤差の低減が見込める場合にのみ，シンクへ送信するセンサデータの追加や除外を行う．その後，センサデータを収集するために割り当てられた時間内にシンクがセンサデータを受信できるように，センサデータを収集する．さらに，提案手法の有効性を示すために，シミュレーション実験による性能評価を行った．その結果より，提案手法は，第 2 章で提案した手法と比較して，推定誤差を維持しつつ，無駄なトラヒックを削減できることを確認した．さらに，境界端末の数が極端に多い環境では，第 2 章で提案した手法より推定誤差が小さくなることを確認した．

イベントの位置と形状を要求するアプリケーションは，収集したセンサデータを使用するアプリケーションの中でも代表的なものであり，これまで数多くの研究が行われている．しかし，その研究のほとんどが，端末の位置が固定されている環境を想定したものであり，これらの既存手法のアプローチを，そのまま端末が移動するモバイル環境には適用できない．これに対し，本論文で提案した手法は，モバイルセンサネットワークにおいて，境界端末の識別およびそのセンサデータの収集に要するトラヒックを削減しつつ，高い精度で境界線を推定できる．特に提案手法は，各端末が事前に隣接端末の情報を所持していない状態においても，効率的に境界線を検出できることから，隣接端末の情報の把握が困難なモバイル環境に適している．本研究の成果により，通信インフラの有無に関わらず，自由に移動する端末のセンサデータを用いて境界線を推定できるようになることから，この研究成果は，1.2 節で例示したようなアプリケーションを，一般ユーザが所持する端末のセンサデータを用いて実現することに向けて，非常に大きな進展をもたらすものである．

5.2 検討課題

本論文における提案手法について，さらなる応用や，実環境への適用を考慮すると，各章のむすびで述べたものに加え，解決しなければならない課題がいくつか残されている．本節では，今後検討する予定である技術課題について述べる．

5.2.1　リアルタイム性の高い境界線のモニタリングへの対応

変化が非常に速いイベントの境界線をモニタリングする場合，アプリケーションは数百ミリ秒程度のごく短い周期で，イベントの位置と形状を要求するものと考えられる．これを実現する方法の一つとして，第 3 章で提案した手法について 3.6 節で述べた今後の課題を解決した後，その手法と第 4 章で提案した手法を組み合わせることで，定期的なクエリのフラッディングをなくしつつ，境界線の形状に応じて，境界端末をサンプリングしながら識別する方法が考えられる．しかしこの方法では，各観測時刻においてシンクへ送信されるセンサデータ数が削減されるとはいえ，そのデータ収集をごく短い周期で繰り返すため，大量のトラヒックが発生してしまう．

一方，文献 [43] で想定しているアプリケーションのように，アプリケーションが境界線の変化をある程度許容する場合，許容できない変化を表すセンサデータのみを収集できれば，ごく小数のセンサデータのみでアプリケーション要求を満足できるものと考えられる．そこで今後は，アプリケーションが境界線の変化をある程度許容する環境において，より効率的に境界線をモニタリングする方法について検討する予定である．

5.2.2　実環境を想定した性能評価

モバイルセンサネットワークやアドホックネットワークを対象とした研究は，提案手法を実システムに実装することが困難であるため，シミュレーション実験による性能評価が行われることが一般的である [12, 21, 24, 31, 45, 47, 55]．また，本論文では，提案手法が有効に機能するか否かにのみ焦点を当てて性能を調べるために，単純なモデルを用いたシミュレーション実験を行った．しかし実環境では，遮蔽物が無線通信に与える影響 [14] など，シミュレーションでは考慮されていない事項が提案手法の性能に大きな影響を与える可能性がある．また，シミュレーション実験時に設定した，タイマや待ち時間を算出する際の定数においても，実環境では異なる値が適切である可能性がある．そのため，より実環境に近い環境を再現したシミュレーション実験を行い，実環境における提案手法の性能を評価

する必要がある.

謝辞

本研究を推進するにあたり，懇切なる御指導と惜しみない御助言を頂きました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 原 隆浩教授に謹んで御礼申し上げます.

本研究を推進するにあたり，直接の御指導，御助言，御討論を頂きました島根大学大学院総合理工学研究科総合理工学専攻 神崎映光准教授に衷心より感謝申し上げます.

本研究において，直接の御指導，御助言，御討論を頂いた大阪大学西尾章治郎総長に深く御礼申し上げます.

本論文をまとめるにあたり，大変有益な御指導と御助言を多数賜りました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 下條真司教授，藤原 融教授に心より感謝申し上げます.

講義，学生生活を通じて，学問に取り組む姿勢をご教授頂きました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　鬼塚　真教授，松下康之教授，萩田紀博教授に厚く感謝申し上げます.

本研究において，多大なる御助言，御協力，御支援を頂きました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 寺西裕一准教授，大阪大学大学院工学研究科 春本 要准教授，大阪大学サイバーメディアセンター 義久智樹准教授，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 前川卓也准教授，天方大地助教，白川真澄特任助教に深謝致します.

本研究において，ともに研究を進め，多大なる御協力を頂いた奈良先端科学技術大学院大学情報科学研究科 後藤啓介氏に深く御礼申し上げます.

本研究を進めるにあたり，多くの御討論や御助言を頂きました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 原研究室の諸氏に心より感謝申し上げます.

最後に，私のこれまでの人生，そして研究生活を送る上で，暖かい支援と理解を頂いた家族や友人に，心から感謝致します.

参考文献

[1] Abolhasan, M., Wysocki, T., and Dutkiewicz, E.: A review of routing protocols for mobile ad hoc networks, *Ad Hoc Networks*, Vol. 2, No. 1, pp. 1–22 (2004).

[2] Alinia, B., Hajiesmaili, M. H., and Khonsari, A.: On the construction of maximum-quality aggregation trees in deadline-constrained WSNs, in *Proc. Int. Conf. on Computer Communications (INFOCOM 2015)*, pp. 226–234 (2015).

[3] Alinia, B., Yousefi, H., Talebi, M. S., and Khonsari, A.: Maximizing quality of aggregation in delay-constrained wireless sensor networks, *IEEE Communications Letters*, Vol. 17, No. 11, pp. 2084–2087 (2013).

[4] Amato, G., Chessa, S., Gennaro, C., and Vairo, C.: Querying moving events in wireless sensor networks, *Pervasive and Mobile Computing*, Vol. 16, pp. 51–75 (2015).

[5] Amundson, I., and Koutsoukos, X. D.: A survey on localization for mobile wireless sensor networks, *Mobile Entity Localization and Tracking in GPS-less Environnments*, pp. 235–254 (2009).

[6] Anastasi, G., Conti, M., Di Francesco, M., and Passarella, A.: Energy conservation in wireless sensor networks: A survey, *Ad Hoc Networks*, Vol. 7, No. 3, pp. 537–568 (2009).

[7] Avci, B., Trajcevski, G., and Scheuermann, P.: Managing evolving shapes in sensor networks, in *Proc. Int. Conf. on Scientific and Statistical Database Management (SSDBM 2014)* (2014).

[8] Bellavista, P., Montanari, R., and Das, S. K.: Mobile social networking middleware: A survey, *Pervasive and Mobile Computing*, Vol. 9, No. 4, pp. 437–453 (2013).

[9] Camp, T., Boleng, J., and Davies, V.: A survey of mobility models for ad hoc network research, *Wireless Communications and Mobile Computing*, No. 5, pp. 483–502.

[10] Campbell, A. T., Eisenman, S. B., Lane, N. D., Miluzzo, E., Peterson, R. A., Lu, H., Zheng, X., Musolesi, M., Fodor, K., and Ahn, G.-S.: The rise of people-centric sensing, *IEEE, Internet Computing*, Vol. 12, No. 4, pp. 12–21 (2008).

[11] Cha, S., Talipov, E., and Cha, H.: Data delivery scheme for intermittently connected mobile sensor networks, *Computer Communications*, Vol. 36, No. 5, pp. 504–519 (2013).

[12] Chen, M., Kwon, T., Yuan, Y., and Leung, V.: Mobile agent based wireless sensor networks, *Journal of Computers*, Vol. 1, No. 1, pp. 14–21 (2006).

[13] Chintalapudi, K. K., and Govindan, R.: Localized edge detection in sensor fields, *Ad Hoc Networks*, Vol. 1, No. 2, pp. 273–291 (2003).

[14] Chipara, O., Hackmann, G., Lu, C., Smart, W. D., and Roman, G.-C.: Practical modeling and prediction of radio coverage of indoor sensor networks, in *Proc. Int. Conf. on Information Processing in Sensor Networks (IPSN 2010)*, pp. 339–349 (2010).

[15] Cisco, T.: Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015–2020, Technical report, Cisco white paper (2016).

[16] Conti, M., and Giordano, S.: Mobile ad hoc networking: Milestones, challenges, and new research directions, *IEEE Communications Magazine*, Vol. 52, No. 1, pp. 85–96 (2014).

[17] Dantu, K., and Sukhatme, G. S.: Detecting and tracking level sets of scalar fields using a robotic sensor network, in *Proc. Int. Conf. on Robotics and Automation Society (RAS 2007)*, pp. 3665–3672 (2007).

[18] Dong, D., Liu, Y., and Liao, X.: Fine-grained boundary recognition in wireless ad hoc and sensor networks by topological methods, in *Proc. Int. Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2009)*, pp. 135–144 (2009).

[19] Duckham, M., Kulik, L., Worboys, M., and Galton, A.: Efficient generation of simple polygons for characterizing the shape of a set of points in the plane, *Pattern Recognition*, Vol. 41, No. 10, pp. 3224–3236 (2008).

[20] Estrin, D.: Participatory sensing: Applications and architecture, *IEEE, Internet Computing*, Vol. 14, No. 1, pp. 12–42 (2010).

[21] Goto, K., Sasaki, Y., Hara, T., and Nishio, S.: Data gathering using mobile agents in dense mobile wireless sensor networks, in *Proc. Int. Conf. on Advances in Mobile Computing and Multimedia (MoMM 2011)*, pp. 58–65 (2011).

[22] Guo, P., Cao, J., and Zhang, K.: Distributed topological convex hull estimation of event region in wireless sensor networks without location information, *IEEE Trans. Parallel and Distributed Systems*, Vol. 26, No. 1, pp. 85–94 (2015).

[23] Guocan, R., and Guowei, D.: An improved isoline based data aggregation scheme in wireless sensor networks, *Procedia Engineering*, Vol. 23, pp. 326–332 (2011).

[24] Gupta, G., and Ramanathan, P.: A distributed algorithm for level set estimation using uncoordinated mobile sensors, in *Proc. Int. Conf. on Global Communications (GLOBECOM 2007)*, pp. 1180–1184 (2007).

[25] Haillot, J., and Guidec, F.: A protocol for content-based communication in disconnected mobile ad hoc networks, *Mobile Information Systems*, Vol. 6, No. 2, pp. 123–154 (2010).

[26] Ham, M. I., and Rodriguez, M. A.: A boundary approximation algorithm for distributed sensor networks, *Journal of Sensor Networks*, Vol. 8, No. 1, pp. 41–46 (2010).

[27] Hanashi, A. M., Awan, I., and Woodward, M. E.: Performance evaluation with different mobility models for dynamic probabilistic flooding in MANETs, *Mobile Information Systems*, Vol. 5, No. 1, pp. 65–80 (2009).

[28] Hariharan, S., and Shroff, N. B.: Maximizing aggregated information in sensor networks under deadline constraints, *IEEE Trans. Automatic Control*, Vol. 56, No. 10, pp. 2369–2380 (2011).

[29] Hasenfratz, D., Saukh, O., Sturzenegger, S., and Thiele, L.: Participatory air pollution monitoring using smartphones, *Mobile Sensing* (2012).

[30] Heissenbüttel, M., Braun, T., Bernoulli, T., and Wälchli, M.: BLR: Beaconless routing algorithm for mobile ad hoc networks, *Computer Communications*, Vol. 27, No. 11, pp. 1076–1086 (2004).

[31] Heo, N., and Varshney, P. K.: A distributed self spreading algorithm for mobile wireless sensor networks., in *Proc. Int. Wireless Communications and Networking Conf. (WCNC 2003)*, Vol. 3, pp. 1597–1602 (2003).

[32] Honicky, R., Brewer, E. A., Paulos, E., and White, R.: N-smarts: Networked suite of mobile atmospheric real-time sensors, in *Proc. ACM SIGCOMM Workshop on Networked Systems for Developing Regions*, pp. 25–30 (2008).

[33] Hsu, W., Dutta, D., and Helmy, A.: CSI: A paradigm for behavior-oriented profile-cast services in mobile networks, *Ad Hoc Networks*, Vol. 10, No. 8, pp. 1586–1602 (2012).

[34] Imran, S., Ercetin, O., and Ko, Y.-B.: Continuous phenomena boundary detection and tracking in wireless sensor networks through optimized monitoring, in *Proc.*

Pacific Rim Conf. on Communications, Computers and Signal Processing (PacRim 2015), pp. 402–406 (2015).

[35] Jabeur, N., Zeadally, S., and Sayed, B.: Mobile social networking applications, *ACM Communications*, Vol. 56, No. 3, pp. 71–79 (2013).

[36] 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).

[37] Jin, G., and Nittel, S.: NED: An efficient noise-tolerant event and event boundary detection algorithm in wireless sensor networks, in *Proc. Int. Conf. on Mobile Data Management (MDM 2006)*, pp. 153–153 (2006).

[38] Jin, G., and Nittel, S.: Efficient tracking of 2D objects with spatiotemporal properties in wireless sensor networks, *Distributed and Parallel Databases*, Vol. 29, No. 1–2, pp. 3–30 (2011).

[39] Jung-Hwan, K., Kee-Bum, K., Chauhdary, S. H., Wencheng, Y., and Myong-Soon, P.: DEMOCO: Energy-efficient detection and monitoring for continuous objects in wireless sensor networks, *IEICE Trans. Communications*, Vol. 91, No. 11, pp. 3648–3656 (2008).

[40] Khan, W. Z., Xiang, Y., Aalsalem, M. Y., and Arshad, Q.: Mobile phone sensing systems: A survey, *IEEE Communications Surveys & Tutorials*, Vol. 15, No. 1, pp. 402–427 (2013).

[41] Kiess, W., and Mauve, M.: A survey on real-world implementations of mobile ad-hoc networks, *Ad Hoc Networks*, Vol. 5, No. 3, pp. 324–339 (2007).

[42] Kim, W., Park, H., Lee, J., and Kim, S.-H.: Efficient continuous object tracking with virtual grid in wireless sensor networks, in *Proc. Vehicular Technology Conf. (VTC 2012)*, pp. 1–5 (2012).

[43] 駒井友香, 原隆浩 : アドホックネットワークにおける凸包領域の継続的な検索手法, データ工学と情報マネジメントに関するフォーラム (DEIM2016) (2016).

[44] Komai, Y., Hara, T., and Nishio, S.: A convex hull query processing method in MANETs, in *Proc. Int. Symposium on Reliable Distributed Systems (SRDS 2014)*, pp. 331–332 (2014).

[45] Komai, Y., Hara, T., and Nishio, S.: Processing convex hull queries in MANETs, in *Proc. Int. Conf. on Mobile Data Management (MDM 2015)*, pp. 64–73 (2015).

[46] Krishnamachari, B., and Iyengar, S.: Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks, *IEEE Trans. Computers*, Vol. 53, No. 3, pp. 241–250 (2004).

[47] Kulik, L., Tanin, E., and Umer, M.: Efficient data collection and selective queries in sensor networks, in *Proc. Int. Conf. on Geosensor Networks*, pp. 25–44 (2006).

[48] Lane, N. D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., and Campbell, A. T.: A survey of mobile phone sensing, *IEEE Communications Magazine*, Vol. 48, No. 9, pp. 140–150 (2010).

[49] Lee, D.-T., and Schachter, B. J.: Two algorithms for constructing a Delaunay triangulation, *International Journal of Computer & Information Sciences*, Vol. 9, No. 3, pp. 219–242 (1980).

[50] Lee, Y. S., Park, J. W., and Barolli, L.: A localization algorithm based on AOA for ad-hoc sensor networks, *Mobile Information Systems*, Vol. 8, No. 1, pp. 61–72 (2012).

[51] Li, F., Luo, J., Zhang, C., Xin, S., and He, Y.: UNFOLD: Uniform fast on-line boundary detection for dynamic 3D wireless sensor networks, in *Proc. Int. Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2011)* (2011).

[52] Li, M., and Liu, Y.: Iso-map: Energy-efficient contour mapping in wireless sensor networks, *IEEE Trans. Knowledge and Data Engineering*, Vol. 22, No. 5, pp. 699–710 (2010).

[53] 松尾和哉, 後藤啓介, 神崎映光, 原隆浩 : 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, 情報処理学会マルチメディア，分散，協調とモバイル (DICOMO2016) シンポジウム論文集, pp. 769–776 (2016).

[54] Matsuo, K., Goto, K., Kanzaki, A., and Hara, T.: A sampling-based boundary detection method with consideration of boundary shape in dense mobile wireless sensor networks, in *Proc. Int. Conf. on Advances in Mobile Computing & Multimedia (MoMM 2016)*, pp. 307–316 (2016).

[55] Matsuo, K., Goto, K., Kanzaki, A., Hara, T., and Nishio, S.: Agent redeployment and missing data regathering in mobile wireless sensor networks, in *Proc. Int. Computer Software and Applications Conf. (COMPSAC 2013)*, pp. 100–109 (2013).

[56] 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を用いた効率的な境界線検出手法, 情報処理学会マルチメディア通信と分散ワークショップ（DPSWS2013）論文集, pp. 236–243 (2013).

[57] Matsuo, K., Goto, K., Kanzaki, A., Hara, T., and Nishio, S.: Overhearing-based efficient boundary detection in dense mobile wireless sensor networks, in *Proc. Int. Conf. on Mobile Data Management (MDM 2014)*, pp. 225–234 (2014).

[58] Matsuo, K., Goto, K., Kanzaki, A., Hara, T., and Nishio, S.: A study on efficient event monitoring in dense mobile wireless sensor networks, in *Proc. Int. Symposium on Reliable Distributed Systems (SRDS 2014)*, pp. 341–342 (2014).

[59] 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).

[60] 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観測値の時間的変化を用いた定期的な境界線検出手法, データ工学と情報マネジメントに関するフォーラム (DEIM2015) (2015).

[61] Meng, X., Nandagopal, T., Li, L., and Lu, S.: Contour maps: Monitoring and diagnosis in sensor networks, *Computer Networks*, Vol. 50, No. 15, pp. 2820–2838 (2006).

[62] 三上岳彦, 大和広明, 安藤晴夫, 横山仁, 山口隆子, 市野美夏, 秋山祐佳里, 石井康一郎：東京都内における夏期の局地的大雨に関する研究, 東京都市環境科学研究所年報, Vol. 2005, pp. 33–42 (2005).

[63] Nowak, R., and Mitra, U.: Boundary estimation in sensor networks: Theory and methods, in *Proc. Int. Conf. on Information Processing in Sensor Networks (IPSN 2003)*, pp. 80–95 (2003).

[64] Park, B., Park, S., Lee, E., and Kim, S.-H.: Detection and tracking of continuous objects for flexibility and reliability in sensor networks, in *Proc. Int. Conf. on Communications (ICC 2010)*, pp. 1–6 (2010).

[65] Park, B., Park, S., Lee, E., Noh, S., and Kim, S.-H.: Large-scale phenomena monitoring scheme in wireless sensor networks, in *Proc. Vehicular Technology Conf. (VTC 2010)*, pp. 1–5 (2010).

[66] Park, S., Hong, S., Lee, E., Kim, S., and Crespi, N.: Large-scale mobile phenomena monitoring with energy-efficiency in wireless sensor networks, *Computer Networks*, Vol. 81, pp. 116–135 (2015).

[67] Reddy, S., Estrin, D., Hansen, M. H., and Srivastava, M. B.: Examining micropayments for participatory sensing data collections, in *Proc. Int. Conf. on Ubiquitous Computing (UbiComp 2010)*, pp. 33–36 (2010).

[68] Reddy, S., Samanta, V., Burke, J., Estrin, D., Hansen, M., and Srivastava, M.: MobiSense - Mobile network services for coordinated participatory sensing, in *Proc. Int. Symposium on Autonomous Decentralized Systems (ISADS 2009)*, pp. 231–236 (2009).

[69] 斎藤洋三, 井手武, 村山貢司：花粉症の科学 (2006).

[70] Serna, M. Á., Bermudez, A., Casado, R., and Kulakowski, P.: A convex hull-based approximation of forest fire shape with distributed wireless sensor networks, in *Proc. Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 2011)*, pp. 419–424 (2011).

[71] Shi, J., Zhang, R., Liu, Y., and Zhang, Y.: Prisense: Privacy-preserving data aggregation in people-centric urban sensing systems, in *Proc. Int. Conf. on Computer Communications (INFOCOM 2010)*, pp. 1–9 (2010).

[72] Singh, A., Nowak, R., and Ramanathan, P.: Active learning for adaptive mobile sensing networks, in *Proc. Int. Conf. on Information Processing in Sensor Networks (IPSN 2006)*, pp. 60–68 (2006).

[73] Solis, I., and Obraczka, K.: Efficient continuous mapping in sensor networks using isolines, in *Proc. Int. Conf. on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous 2005)*, pp. 325–332 (2005).

[74] 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).

[75] Srinivasan, S., Dattagupta, S., Kulkarni, P., and Ramamritham, K.: A survey of sensory data boundary estimation, covering and tracking techniques using collab-

orating sensors, *Pervasive and Mobile Computing*, Vol. 8, No. 3, pp. 358–375 (2012).

[76] Srinivasan, S., and Ramamritham, K.: Contour estimation using collaborating mobile sensors, in *Proc. Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks*, pp. 73–82 (2006).

[77] 高橋一之, 高橋日出男：夏季夜間の局地風系に及ぼす東京都心域のヒートアイランド現象の影響: 高密度観測網の気圧データを用いた事例解析, 天気, Vol. 60, No. 7, pp. 505–519 (2013).

[78] Tseng, Y.-C., Ni, S.-Y., Chen, Y.-S., and Sheu, J.-P.: The broadcast storm problem in a mobile ad hoc network, *Wireless Networks*, Vol. 8, No. 2-3, pp. 153–167 (2002).

[79] 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).

[80] Wang, B.: Coverage problems in sensor networks: A survey, *ACM Computing Surveys*, Vol. 43, No. 4, pp. 32:1–32:53 (2011).

[81] Wang, Y., Gao, J., and Mitchell, J. S.: Boundary recognition in sensor networks by topological methods, in *Proc. Int. Conf. on Mobile computing and networking (MobiCom 2006)*, pp. 122–133 (2006).

[82] Worboys, M., and Duckham, M.: Monitoring qualitative spatiotemporal change for geosensor networks, *Journal of Geographical Information Science*, Vol. 20, No. 10, pp. 1087–1108 (2006).

[83] Yang, Y., Liu, Y., Zhang, Q., and Ni, L.: Cooperative boundary detection for spectrum sensing using dedicated wireless sensor networks, in *Proc. Int. Conf. on Computer Communications (INFOCOM 2010)*, pp. 1–9 (2010).

[84] Yick, J., Mukherjee, B., and Ghosal, D.: Wireless sensor network survey, *Computer Networks*, Vol. 52, No. 12, pp. 2292–2330 (2008).

[85] Zandbergen, P. A.: Positional accuracy of spatial data: Non-normal distributions and a critique of the national standard for spatial data accuracy, *Trans. GIS*, Vol. 12, No. 1, pp. 103–130 (2008).

[86] Zandbergen, P. A.: Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning, *Trans. GIS*, Vol. 13, No. 1, pp. 5–25 (2009).

[87] Zandbergen, P. A., and Barbeau, S. J.: Positional accuracy of assisted GPS data from high-sensitivity GPS-enabled mobile phones, *Journal of Navigation*, Vol. 64, No. 3, pp. 381–399 (2011).

[88] Zhong, C., and Worboys, M.: Energy-efficient continuous boundary monitoring in sensor networks, *National center for Geographic Information and Analysis* (2007).

[89] Zhong, C., and Worboys, M.: Continuous contour mapping in sensor networks, in *Proc. Int. Consumer Communications and Networking Conf.*, pp. 152–156 (2008).

[90] Zhou, H., Wu, H., and Jin, M.: A robust boundary detection algorithm based on connectivity only for 3D wireless sensor networks, in *Proc. Int. Conf. on Computer Communications (INFOCOM 2012)*, pp. 1602–1610 (2012).

[91] Zhu, C., Shu, L., Hara, T., Wang, L., Nishio, S., and Yang, L. T.: A survey on communication and data management issues in mobile sensor networks, *Wireless Communications and Mobile Computing*, Vol. 14, No. 1, pp. 19–36 (2014).

[92] Zhu, X., Sarkar, R., Gao, J., and Mitchell, J. S.: Light-weight contour tracking in wireless sensor networks, in *Proc. Int. Conf. on Computer Communications (INFOCOM 2008)*, pp. 1175–1183 (2008).