

センサネットワークにおける エネルギー消費の少ないトラッキングアルゴリズム

植村 奈緒子†

大下 福仁†

角川 裕次†

増澤 利光†

† 大阪大学大学院情報科学研究科

1 はじめに

センサネットワークとは、多数の小型センサ端末(センサノード)が、無線通信を行うことにより構成されるネットワークである。センサネットワーク上のアプリケーションは観測領域のイベント(e.g. 温度の変化、火災の発生、物体の移動)を検知し、ユーザにイベントの情報を知らせる。このとき、各センサノードはバッテリーを用いて動作するため、センサネットワークにおいては、エネルギー消費を抑えることが非常に重要である。

センサネットワークを用いた重要なアプリケーションの1つが、移動体の位置情報や移動の軌跡を取得するトラッキングである。トラッキングは、例えば、戦場での敵兵検知や、遠隔地域での野生動物の監視などに利用される。より正確には、トラッキングとは移動体の検出情報(e.g. 位置、センシングの計測値等)をユーザが取得することである。以降、観測対象の移動体を目標物と呼ぶ。ユーザは、クエリと呼ばれるメッセージを用いて検出情報を要求する。目標物の情報を得るために、ユーザはまず、センサネットワーク内のセンサノードを経由してクエリを転送させる。そして、目標物の情報を持つセンサノードが、受信したクエリに応じて目標物の情報を返送する。しかしセンサノードは、送受信を行うときに多くのエネルギーを使用する。このため、効率的にクエリや目標物の情報を転送させることで、トラッキングにおけるエネルギーを節約することが重要である。

多くの既存研究では、効率的に目標物の情報を得るためにシンクノードを利用している[1]。シンクノードとは、情報を保管しておくためのノードである。シンクノードとしてセンサネットワーク内の1センサノードを選び、情報保管の役割を持たせる。シンクノードを利用した手法では、センサネットワーク構築時に、あらかじめ全センサノードにシンクノードの情報を知らせておく。各センサは、目標物の情報を検出するたびに、シンクノードにデータを送信し蓄積する。ユーザはシ

ンクノードにクエリを送信することで、ネットワーク中にクエリを広めることなくデータを引き出すことが可能である。

しかし、シンクノードを利用した手法では、ユーザの要求に関わらずシンクノードへデータを蓄積するため、ユーザが必要としない期間にもエネルギーを消費し続けてしまう。また、ユーザの位置がシンクノードから遠い場合にはメッセージのホップ回数が増加するため、エネルギーの消費が多くなってしまう。

本稿では、シンクノードを利用しないことによりエネルギー消費の少ないトラッキングアルゴリズムを提案する。

2 諸定義

全てのセンサノードは、等しい大きさの検出領域を持つ。各センサノードは、自身の検出領域内に存在する目標物を検出し、目標物の位置情報を計測可能とする。センサノード v がセンサノード u の検出領域と共通部分を持つとき、センサノード u をセンサノード v の隣接ノードと呼ぶ。センサノード v の隣接ノードの集合を N_v と書く。全てのセンサノードは、同一の通信半径を持ち、通信半径内のセンサノードと通信可能とする。ただし、通信半径は検出領域よりも十分大きいとし、隣接ノードとは必ず通信可能とする。また、各センサノードはGPS等を用いて、自身の位置情報を保持できるとする。

ユーザは観測領域を自由に移動する。このため、ユーザはセンサネットワーク内の任意のセンサノードからクエリを送信すると考える。本稿では、ユーザ、目標物はそれぞれ1つずつの場合のみを考える。また、観測領域内のどの地点も、少なくとも1つのセンサノードの検出領域に含まれているとする。

3 提案手法

3.1 概要

各センサノード v は目標物が自身の検出領域から出たときに、そのとき目標物を検出しているセンサノード $u \in N_v$ を推定する。このノードを移動先ノードと呼

†Naoko Uemura Fukuhito Ooshita Hirosugu Kakugawa Toshimitsu Masuzawa

†Graduate School of Information Science and Technology, Osaka University

ぶ。各ノードの保持する移動先ノードを辿ることで、時刻 t に目標物を検出したセンサノード v から時刻 $t' > t$ に目標物を検出したセンサノード u への経路 (移動経路と呼ぶ) を得ることができる。

ユーザは前回クエリに返答したセンサノードの情報を保持しておく。このセンサノードを前任ノードと呼ぶ。ユーザはこの前任ノードにクエリを送信する。センサネットワーク内の各センサノードはクエリを受信したとき、自身の保持する移動先ノードへクエリを転送する。クエリの転送を繰り返すことで、現在目標物を検出しているセンサノード (検知ノード) にクエリを渡すことができる。

3.2 移動先ノードの取得

各センサノードは取得した目標物の位置情報を基に、隣接ノードの中から移動先ノードを選択する。以降では、この移動先ノードの選択方法について説明する。

各センサノード v は目標物が自身の検出領域から出たときに、自身が最後に目標を検出した位置 (最終検出位置) と全隣接ノードとの距離を比較する。ある隣接ノード $u \in N_v$ と最終検出位置との距離が検出半径 r 以下ならば、隣接ノード u は目標物を必ず検出していることが分かる。このため、 v はセンサノード u を移動先ノードとして記憶する。図 1 では、この後目標物がセンサノード v の検出領域から出たならば、センサノード u を移動先ノードとして記憶する。

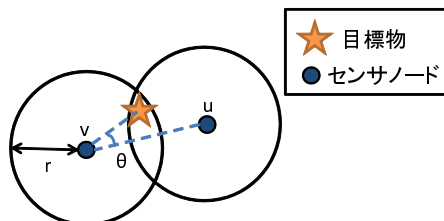


図 1: 移動先ノードの取得

移動先ノードの候補が複数存在する場合は、最終検出位置の方向を基に移動先ノードを決定する。センサノード v は、移動先ノードの候補 $u \in N_v$ と最終検出位置とが作る角 θ を計算する。最も角 θ が小さい隣接ノードを移動先ノードとする。これにより、目標物が移動している方向のノードを移動先ノードとして選ぶことができる。

3.3 近道の作成

提案手法では、目標物が移動するにつれて移動経路が長くなるため、ユーザが情報を得るまでに遅延が生

じてしまう。これを解決するために、一定時間ごとに近道の作成 (経路の圧縮) を行う。詳細は省略する。

4 シミュレーション

提案手法の有効性を確認するために、既存研究 [1] で提案されている EASE との比較シミュレーションを行った。

まず、シミュレーションの設定を説明する。ネットワークは、 $480 \times 680m^2$ の観測領域を 1 セル $40 \times 40m^2$ の格子状に分割し、各セルの中心にセンサノードを配置した。各センサの持つ検出領域は自身を含む 1 セルを覆う円形とする。すなわち、直径 $40\sqrt{2}$ の円である。通信半径は 40m とする。目標物の移動モデルとしてはランダムウェイポイントモデルを利用した。ユーザのクエリ送信頻度は 1.0 回/秒とする。

図 2 に、3 種類の移動速度を用いた場合の各手法によるメッセージ数の比較を示す。提案手法は EASE よりも 35 ~ 55% 程度メッセージ数を削減していることが分かる。

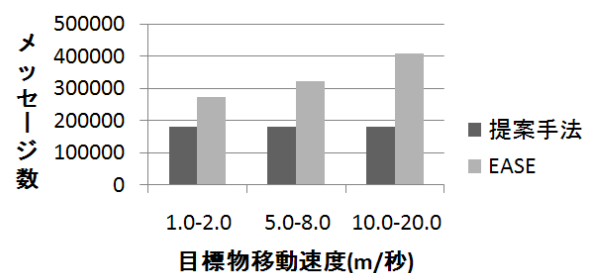


図 2: メッセージ数による比較

5 まとめ

本稿では既存のトラッキングアルゴリズムの問題点について説明し、エネルギー消費の少ないトラッキングアルゴリズムを提案した。また、シミュレーションを用いて、既存のシンクノードを利用する手法よりもメッセージ数を削減できることを示した。

参考文献

- [1] Jianliang Xu, Xueyan Tang, and Wang-Chien Lee. A new storage scheme for approximate location queries in object-tracking sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 19, No. 2, p. 262, 2008.