



移動型センサネットワークにおける複数ノードのための経路探索手法

中宮, 正樹
寺田, 努
西尾, 章治郎

(Citation)

情報処理学会論文誌, 50(4):1262-1271

(Issue Date)

2009-04-15

(Resource Type)

journal article

(Version)

Version of Record

(URL)

<https://hdl.handle.net/20.500.14094/90002110>



移動型センサネットワークにおける 複数ノードのための経路探索手法

中 宮 正 樹^{†1} 寺 田 努^{†2} 西 尾 章 治 郎^{†3}

筆者らはこれまで、従来の移動型センサネットワークの研究では扱われていなかった実用上の問題を解決するために、コストマップを用いた移動型センサノードのための経路探索手法を提案した。提案手法により、センサのセンシング範囲やノード移動時の障害物、ノードの移動特性を考慮したノードの経路探索が可能となった。本稿では新たに、複数ノードどうしの衝突を考慮したアルゴリズムと、センシング領域を指定するための問合せ言語を提案する。これらの手法により、ユーザは複数のノードを用いて複数のセンシング領域を効率良くセンシングできるようになる。さらに、使用するノードの移動特性や搭載されているセンサの種類や数、センシング範囲に関する知識がなくても、問合せ言語を記述することで複数のノードがセンシング要求に応じた最適な場所へ自動的に配置されるようになる。本稿では、複数ノードどうしの衝突を回避するアルゴリズムの有効性を確認するためにシミュレーションによる評価を行った。

A Route Planning Method for Multiple Mobile Sensor Nodes

MASAKI NAKAMIYA,^{†1} TSUTOMU TERADA^{†2}
and SHOJIRO NISHIO^{†3}

We have proposed a route planning method for a mobile sensor node using *cost map*. The proposed method achieves a novel path planning that can solve several practical problems in conventional work: limitations of sensing areas, barricades on nodes' path, and restrictions on nodes' movements. In this research, we propose a new route planning method that can avoid collisions based on the previous method. Moreover, we also propose a query language to specify required sensing data. The proposed method effectively allocates multiple sensor nodes. We show the effectiveness of the proposed method through simulation study.

1. はじめに

近年、無線通信技術とネットワーク技術の発展やデバイスの小型化の恩恵を受けて、センサデバイスに無線通信機能をもたせたセンサノードによってネットワークが形成される、センサネットワークが実現しつつある¹⁾。さらに最近では、センサノードにアクチュエータを搭載し、自由に移動できるようにした移動型センサノードに関する研究が注目されている^{3),7)}。ノードに移動性を持たせることで、センシングした結果に応じて次にセンシングする位置を決定するなどセンシングデータ要求の動的な変化に対応でき、人が近づけない危険地帯の探索などが可能となる。

筆者らはこれまでに、多種の移動モデルや多種のセンシングモデルをもつ移動型センサネットワークを統合的に扱うために、コストマップを用いた移動型センサノードの経路探索手法を提案した⁵⁾。提案手法により、センサの種類によるセンシング可能な範囲の違い、ノード移動時の障害物、およびノードの移動特性を考慮し、ノードが移動に要する消費電力が最小となる経路を探索することが可能となった。

しかし、これまでの手法では単一のノードのみを想定していたため、複数のノードを同時に使用する際にはノードどうしの衝突が起こるという問題があった。そこで本稿ではまず、複数ノードどうしの衝突を考慮した経路探索アルゴリズムを提案する。これにより、複数ノードを複数箇所のセンシング領域へ低コストで配置できるようになる。また、従来手法では、移動型ノードを用いてセンシングを行うために、ユーザはノードに搭載されているセンサの種類やセンサデータの必要な時刻などを考慮して、どのノードをどのセンシング領域へ移動させるのかを指定しなければならなかった。そこで、移動型ノードを用いたセンサネットワークのための問合せ言語を提案する。これによりユーザは、センシングしたい場所や時刻、センサの種類といったセンシング条件を容易に設定でき、さらに使用するノードの数や移動特性、搭載されているセンサの種類や特徴に関する詳細な知識がなくても複数ノードをセンシングの目的に応じて低コストで配置できるようになる。

本研究では、ノードどうしの衝突を考慮したアルゴリズムに関して、配置されるまでに

^{†1} 株式会社 NTT データ

NTT Data Corporation

^{†2} 神戸大学大学院工学研究科

Graduate School of Engineering, Kobe University

^{†3} 大阪大学大学院情報科学研究科

Graduate School of Information Science and Technology, Osaka University

ノードが移動に要する電力量および全ノードの配置が完了するまでの消費時間の比較を行い、それぞれが他のアルゴリズムと比べて有効となる状況を示した。

以下、2章で想定する環境について述べ、3章で提案手法について説明する。4章でシミュレーションによる性能評価について述べ、5章で考察を行い、6章で結論と今後の課題を述べる。

2. 想定環境

本研究では、ビル内や地下鉄の構内などの地形が変化しない静的な環境におけるセンシングを想定し、地形の情報は図1に示すような2次元格子状に区切られたフィールドマップとしてあらかじめ与えられているものとする。フィールドマップの1つの格子を本研究ではセルと呼び、セルごとに、凹凸の多い地面や障害物といったノードの移動に影響を及ぼす情報が、地形コストとして設定されている。このような状況において、ノードの移動を制御するサーバに対して、ユーザから単一ノードではまかなえないような複数箇所での広範囲にわたるセンシングデータの要求があり、要求に対して多種のノードを複数台使用して協調的にセンシングを行う。なお、移動型ノードを用いたセンサネットワークではノードが移動する際の消費電力を低く抑えることが重要であることから、本研究ではノードの消費電力をコストとし、コストの削減を目的とする。

移動型センサネットワークで使用するノードの種類としては、車型ノードや2足歩行型ノードなど様々な種類が考えられる。それぞれ移動方法に特性をもち有利な環境が異なるため、移動型センサネットワークでは同じ種類のノードを複数利用せず、様々な種類のノードを併用してセンシングすることが望ましい。また、最近は様々なセンサが日常生活で利用されており、センサネットワークにおいても、ユーザが知りたい情報により使用するセンサが

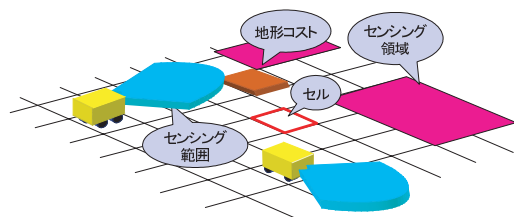


図1 フィールドマップ
Fig.1 Field map.

異なるため、ノードに多種のセンサを搭載することが望ましい。このような観点から、本研究では様々なセンサを搭載した多種のノードを併用したセンシングを行う環境を想定する。

さらに、本研究ではノードはスタート地点からセンシング領域へ効率良く移動することを研究対象とし、移動後のセンシング時間・頻度やそのセンシングデータの転送に特定の方式を仮定しない。また、対象エリアに到着後は到着地でそのままセンシングを継続し続けることは少なく、次のセンシングエリアへ移動したりスタート地点へ戻ったりするなどの処理を行い、移動のコストを下げることでトータルのコストを下げることに繋がる環境を想定する。

3. 提案手法

移動型ノードを用いたセンサネットワークの利用目的は様々であり、ユーザの目的に応じて要求されるセンサデータが異なる。要求されるセンサデータには、センシングしたい範囲やセンサの種類、センシングする時刻やセンシングの精度など複数の条件が同時に設定されると考えられる。ユーザがセンシングの目的に応じて移動型ノードを利用するためには、これらの条件を容易に指定できる必要がある。

また、本研究で想定するように、各ノードが複数のセンサを搭載し様々な種類のノードを併用して複数箇所のセンシングを行う場合、ノードの配置パターンが複数存在する。たとえば、ある地点における温度、湿度、赤外線の種類3種類のセンシング要求に対し、3台のノードがそれぞれ（湿度、温度）、（温度、赤外線）、（赤外線、湿度）の2つずつのセンサを搭載していた場合、ノードの配置パターンは3通りある。そのため、ユーザからの要求をもとに各ノードの移動経路探索を行ったうえで合計コストが最小となる配置パターンを決定しなければならない。そこで本稿では、これらの要件を満たす、複数ノードを用いたセンサネットワークのための問合せ言語とノードどうしの衝突を考慮した経路探索アルゴリズムを提案する。

3.1 問合せ言語

従来、センサネットワーク向けの問合せ記述言語としては、TinyDB⁴⁾をはじめとしていくつかの方式が存在する。一方、これら既存の問合せ言語は固定型ノードからなるセンサネットワークのための問合せ言語であり、本研究で対象とする移動型センサネットワークのための問合せでは、「どこまでセンシング目的に近づく必要があるか」といった要件を求めるセンサデータの精度に基づいて記述することが重要になる。

そこで本研究では、SQL (Structured Query Language) をもとにした移動型センサネッ

トワーク向けの問合せ言語を提案する．ユーザは必要なセンサデータの条件として，センシング領域，センサの種類，制限時間，精度を指定する．センシング領域とは，ユーザがセンサデータを取得したい場所であり，長方形で表現する．センサの種類とは，ユーザが欲しいセンサデータの種類を表し，同じ種類のセンサを搭載したノードがセンシング領域へ配置される．制限時間は，ユーザがセンサデータを取得したい時間を意味し，本稿ではノードがこの時間内に到達すればユーザはデータを取得できるものとしている．以下に問合せの構文を示す．

```
SELECT  type, accuracy
FROM    (x, y), (width, height)
LIMIT  time
```

まず，SELECT 文でセンサの種類 *type* と精度 *accuracy* を指定する．*type* では，*temp* で温度センサ，*humid* で湿度センサといったセンサの種類を指定する．*accuracy* ではセンサのセンシング範囲がセンシング領域を完全に包含するかしないかといった形で指定することができる．具体的には *accuracy* は数値で指定し，1 であればノードはセンサのセンシング範囲がセンシング領域を包含する配置場所，0 であればセンサのセンシング範囲が一部センシング領域と被る配置場所の中から，移動コストが最小のものがそれぞれ選択される．次に，FROM 文でセンシングしたい範囲を座標で指定する． (x, y) でセンシング領域の最左上座標を指定し，*width* でセンシング領域の横のセル数，*height* でセンシング領域の縦のセル数を指定する．さらに，LIMIT 文でセンシング領域ごとの制限時間 *time* を指定でき，ノードは制限時間内にセンシング領域へ配置される．なお *time* は，指定がなければ無制限とする．

3.2 衝突を考慮した経路探索アルゴリズム

問合せ言語を用いてユーザが記述したセンサデータの要求をもとに，衝突を考慮して複数ノードをセンシング領域へ配置させる．

提案手法の手順

提案手法では以下の順序でノードの移動経路を決定する．

Step 1: ユーザが指定したセンサデータの種類とノードに搭載されているセンサの種類を比較し，それらが一致するセンシング領域とノードの組合せをすべて列挙する．

Step 2: 次に，1 つのセンシング領域に関して，ノードが搭載するセンサのセンシング範囲とセンシング領域が重なるようなノードの配置位置および向きを，すべて列挙する．センシング領域ごとにスタート地点から候補の配置位置までの経路探索を従来の経路探索アル

3	2	2	2	3
5	2	1	2	5
5	6		6	5
5	2	1	2	5
3	2	2	2	3

(a) ベースコスト A

(ノード 90° 方向)

(a) Base cost A.

6	4	2	2	2
4	3	1	1	2
2	1		1	2
2	1	1	3	4
2	2	2	4	6

(b) ベースコスト B

(ノード 45° 方向)

(b) Base cost B.

図 2 車型ベースコスト

Fig. 2 Base cost for car type node that direction of node is not considered.

ゴリズムを用いて行い，コストが最小となる 1 つの経路を決定する．この際，ノードどうしの衝突を考慮するために *Score* だけでなく，ノードがセルを通過する時間 *Time* も同時に計算する．

Step 3: センシング領域とノードのすべての組合せの中で合計コストが最も小さくなる組合せを選択する．

Step 4: ノードがセルを通過する時間を比較してノードどうしの衝突を検出する．ノードが 2 セル移動する時間内に，周囲 1 セル内に複数のノードが存在する状態を衝突とする．ノードにはあらかじめ優先度がつけられており，衝突を回避するために優先度がより低いノードについて提案手法による経路の再探索を行う．

Step 5: 最終的に，合計コストが最も小さくなる組合せを選択する．

衝突を考慮した経路探索アルゴリズム

提案手法は，我々がこれまでに提案したコストマップを用いた経路探索アルゴリズムをもとにしている．まず従来手法について簡単に説明し，提案手法について従来手法との違いを述べながら説明する．従来手法では，経路探索を行う前にノードが地形の影響がない状況で周辺のセルへ移動するのに要するベースコスト (図 2) をあらかじめ測定する．測定は，移動後のノードの向きによるコストの違いを考慮して 8 方向別に行う．なお，図 2 では説明の簡単化のため向きは考慮していない．セルごとに以下の簡単な計算により *Score* を求め，コストマップを作成することでコストが最小となる経路の探索を行う．

$$Score = C + H$$

C は，スタート地点から注目するセルまでの移動に要する総移動コスト，*H* は，注目するセ



図3 フィールドマップ一例
Fig. 3 Example of field map.

ルからゴール地点まで地形コストがないと仮定した場合の移動コストである。Score はノードの向きにより 8 方向で区別し、1 つのセルに対し 8 方向別に Score を計算する。Score を計算する際には、ノードの向きによって図 2(a) に示す 90° 方向と図 2(b) に示す 45° 方向のベースコストを使い分け、ノードの向きに合わせて計算する。また、経路探索の過程ではセルは以下の 3 つの状態に分類される。

未計算: Score の計算が行われていないセル

OPEN: Score の計算が行われたセル

CLOSE: Score の計算が行われ、そのセルまでの最短経路が確定しているセル

最初に、スタート地点を中心としたベースコストの範囲内に含まれるすべてのセルについて Score の計算を行い、これらのセルを OPEN 状態とする。次に、OPEN 状態のセルの中で最小 Score をもつセルを CLOSE 状態にし、このセルを中心としたベースコストの範囲内に含まれるすべてのセルについて Score を計算する。以降、ゴール地点が CLOSE 状態となるまで計算を繰り返すことで経路探索を行う。また、各セルには自身の Score が計算される直前に注目されていたセルの位置と、そのセルからの移動経路が記録されているため、ゴール地点からスタート地点までをたどると最小コスト経路が得られる。

提案手法では Score の計算と同時にスタート地点からの移動に要する時間も計算し、Time と表記する。経路探索を行うと同時に他ノードが通過する時間を比較する。一定時間内に周囲 1 セル以内に複数ノードが存在する状況を衝突とし、回避するために直前で停止するまたは他の経路を探索する。このとき、ノードは停止したり再度動き始めたりする際にも電力を消費する。

ここで、図 3 に示すフィールドマップと、図 2 に示すベースコスト、および図 4 を用いて、複数ノードの衝突を考慮した経路探索アルゴリズムの手順を説明する。なお、この例で

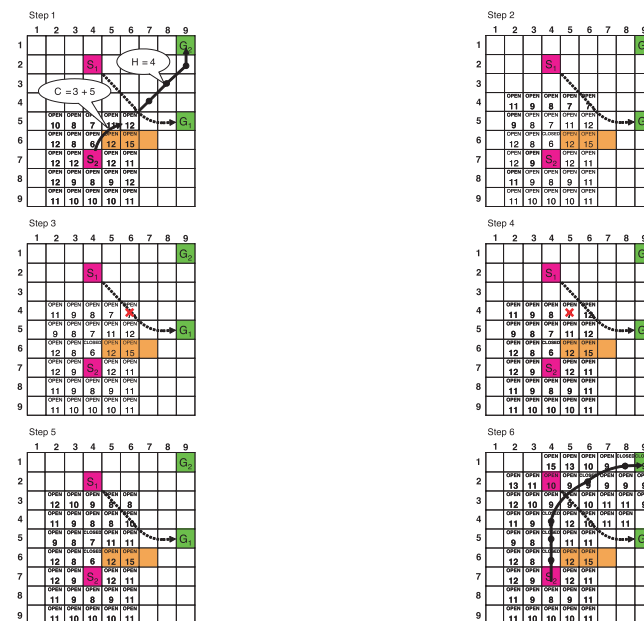


図4 経路探索アルゴリズム適用例
Fig. 4 Example of route planning.

は説明の簡単化のため移動後のノードの向きによるコストの違いを考慮しない。また、優先度の最も高いノードの経路探索がすでに終了しているものとする。以降の説明ではセルを特定するために、フィールドマップの列番号 x 、および行番号 y を用いてセルの位置を (x, y) のように表す。たとえば、図 3 に示すフィールドマップにおいてスタート地点 2 の位置は $(4, 7)$ 、ゴール地点 2 の位置は $(9, 1)$ となる。

Step 1: スタート地点から、ベースコストの範囲に含まれるセルの Score および Time を計算して OPEN 状態にする。スタート地点でのノードの向きが上向きであるため Score の計算にはベースコスト A を用いる。たとえば、セル $(6, 5)$ では C はベースコストによるコスト 3 に途中通過する地形コスト 5 を加えた 8 となり、 H は 4 となるため、Score は 12 となる。

Step 2: OPEN 状態のセルの中で Score が最小となるセル $(4, 6)$ に注目し、CLOSE 状態にする。注目したセル $(4, 6)$ から、ベースコストの範囲に含まれるセルの Score を計算

して OPEN 状態にする．セル (4, 6) でのノードの向きが上向きであるため *Score* の計算にはベースコスト A を用いる．このとき OPEN 状態にされたセルには，直前に注目されていたセルの位置と，そのセルからの移動経路および *Time* を記録しておく．ここで，セル (2, 8)，(3, 7)，(6, 8) ではセル (4, 6) から計算された *Score* の方が小さいため，*Score* と移動経路および *Time* を更新する．

Step 3: 以降，同様の手順を繰り返すと，セル (6, 4) のように優先度の高いノードが通過するセルに注目する場合がある．ここで，記録されている通過時間と経路探索中のノードがそのセルを通過する時間 *Time* を比較し，時間差がノードが 2 セル分の移動に要する時間以下であれば衝突するものとみなす．衝突がないと判断した場合は Step 5 に進む．

Step 4: 衝突するとみなした場合，それまで探索してきた経路は使用できないため他の経路を探索する．また，他のノードと衝突しなくなるまで衝突する直前のセルで停止した場合の経路も同時に探索する．後者の経路を探索するために，再び動き始めてセル (6, 4) へ移動した場合の *Score* および *Time* を計算し，新しい情報として更新する．なお，ノードは停止および動き出しに電力を余分に消費する．次に，OPEN 状態の全セルの中で *Score* が最小となるセル (5, 4) に注目する．提案手法では，優先度の高いノードが通過するセルに隣接するセルについても同様に衝突の判定を行う．この例ではセル (5, 4) についても，記録されている通過時間と経路探索中のノードがそのセルを通過する時間 *Time* を比較した結果衝突するとみなされ，セル (6, 4) と同様に直前のセルで停止した場合の *Score* および *Time* を新しい情報として更新する．

Step 5: OPEN 状態の全セルの中で *Score* が最小となるセル (4, 5) に注目して CLOSE 状態にし，ベースコストの範囲に含まれるセルの *Score* と *Time* を計算して OPEN 状態にする．セル (4, 5) でのノードの向きが上向きであるため *Score* の計算にはベースコスト A を用いる．

Step 6: ゴール地点が注目されて CLOSE 状態となるまで *Score* の計算を繰り返し行う．セルには，自身の *Score* が計算される直前に注目されていたセルの位置と，そのセルからの移動経路が記録されているため，ゴール地点からスタート地点までをたどることで，ノードどうしの衝突を回避する経路が得られる．この例では途中，優先度の高いノードの経路と交わっているが，そのセルを通過する時間に差があるため衝突しない．

4. 性能評価

提案するアルゴリズムの性能を評価するためのシミュレーションを行った．地形コスト



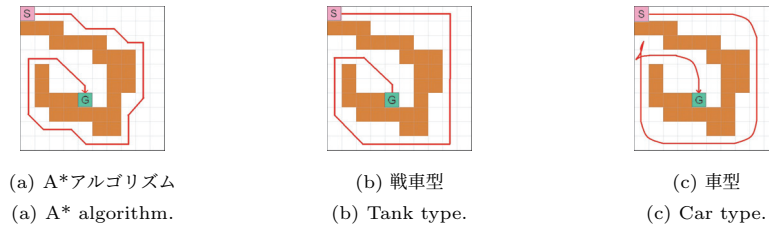
図 5 プロトタイプ
Fig. 5 Prototypes of sensor nodes.

は，壁などのノードが通過できない障害物を想定し，コストが無限大のもののみを配置した．ノードは，前進，後退，旋回を行う戦車型ノードと，前輪で角度をつけて前進，後退を行う車型ノードの 2 種類を考える．実際に図 5 に示す LEGO 社 MindStorm¹²⁾ を用いてこれら 2 種類のノードを製作し，3×3セルのベースコストをあらかじめ測定して経路探索を行った．フィールドマップにおける 1つのセルの大きさは実際に作成したノードの大きさ相当とし，20 cm 四方とする．

4.1 従来手法における経路探索結果の比較

筆者らはこれまでに，ノードの移動特性および経路上の障害物を考慮した最小コスト経路探索アルゴリズムを提案し，人工知能の分野で経路探索に広く使用されている A*アルゴリズムとの比較を行った⁵⁾．A*アルゴリズムは，途中で通過する経路のコストを考慮して最短経路の探索を行うためのアルゴリズムである．A*アルゴリズムでは，経験的に最終状態に近づく経路を優先的に探索していくことで計算量を軽減させ，解となる経路がある場合には必ず最短経路が得られることが証明されている¹⁾．

A*アルゴリズム，戦車型ノードを想定した提案アルゴリズム，および車型ノードを想定した提案アルゴリズムによる経路探索結果の一例，およびその際のベースコストの内容を図 6 に示す．ベースコストは 3×3 の場合であり，それぞれのセルは 8 方向分のコストを値としてもっている．A*アルゴリズム (図 6 (a)) はゴール地点までの最短経路を求めるが，最短経路が複数ある場合，ゴール地点により早く近づく経路を優先的に選択するため，このように方向転換する回数が多い移動経路が得られる場合がある．提案アルゴリズムで戦車型ノードを想定した結果 (図 6 (b)) では A*アルゴリズムの結果と比べて，方向転換の回数が少ない経路が探索されていることが分かる．これは，提案アルゴリズムでは経路探索前に実際のベースコストを測定しており，方向転換に消費する電力を考慮した経路探索を行



(S: 初期位置, G: 目的位置)

5.3	7.2	7.7	5.0	3.1	5.0	7.7	7.2	5.3
7.2	8.3	6.5	5.5	8.3	7.2			
7.7	8.3	8.4	6.0	6.2	6.0	8.4	8.3	7.7
7.0	7.5	7.0	1.9	0	1.9	7.0	7.5	7.0
5.2	5.2	2.4	2.4	5.2	5.2			
7.0	7.5	7.0	2.9	3.4	2.9	7.0	7.5	7.0
7.7	7.2	5.4	5.0	3.1	5.0	5.4	7.2	7.7
8.3	7.2	5.5	5.5	7.2	8.3			
8.4	8.3	7.7	6.0	6.2	6.0	7.7	8.3	8.4

戦車型のベースコスト

Base cost for tank type.

2.2	5.9	6.9	4.9	1.2	4.9	6.9	5.9	2.2
4.3	6.0	5.1	5.1	6.0	4.3			
7.2	8.6	7.3	6.1	7.0	6.1	7.3	8.6	7.2
5.5	7.5	3.4	3.7	0	3.7	3.4	7.5	5.5
5.5	7.4	3.5	3.5	7.4	5.5			
9.6	9.5	10.0	7.7	7.9	7.7	10.0	9.5	9.6
6.5	5.9	4.9	1.8	1.2	1.8	4.9	5.9	6.5
3.9	4.3	5.1	5.1	4.3	3.9			
11.6	10.2	8.0	5.3	8.7	5.3	8.0	10.2	11.6

車型のベースコスト

Base cost for car type.

図 6 経路探索結果例

Fig. 6 Results of route planning.

うためである。車型ノードを想定した場合（図 6(c)），広い道での曲がり角では，緩やかな曲線を描きながら 90° 右方向へ曲がっているが，狭い道では，いったん曲がり角を通過して停車後，後退しながら進行方向に向き，前進するという経路が選択されている。これは，車型ノードが進行方向から見て 90° 方向へは移動できないという移動特性が考慮されていることを意味する。

なお，この比較における消費電力改善率や他の環境における評価など，単一ノードを対象としたコストマップ利用の効果については文献 5) に詳細を示している。

4.2 衝突回避アルゴリズムの評価

提案した衝突を考慮した経路探索アルゴリズムについて，ノードのスタート時間を遅らせることでノードどうしの衝突を回避するアルゴリズム，および衝突する直前で停止することで衝突を回避するアルゴリズムとの比較評価を行った。スタート遅延手法は，まず各ノードの目的地となりうるすべての地点までの経路を探索し，その中で移動コストが最小となる配置場所を求める。次に，すべてのノードがセンシング領域に配置される組合せを列挙し，

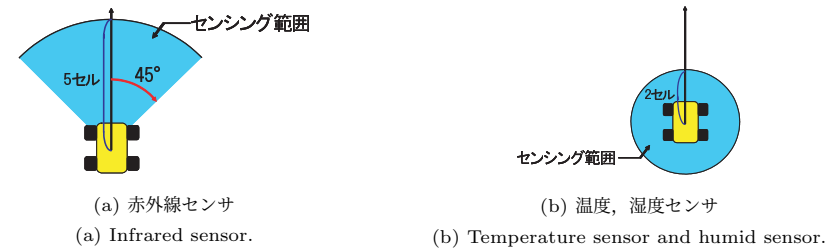


図 7 センシングモデル

Fig. 7 Model of sensing area.

全ノードの移動コストの合計が最小となる組合せを選択する。衝突が発生する場合，衝突を避けるために優先度の低いノードは衝突が起こらなくなるまでスタート時間を遅らせる。以降，全ノードについて優先度の高いノードと衝突しないでセンシング領域に到達するまで処理を繰り返す。ノードはスタート地点での待機も含めて停止中も待機電力を消費する。待機電力は，実機により測定した値を用いる。なお，待機電力量は同じ時間移動するのに消費する電力量の約半分であった。

評価指標としては，全ノードが同時に移動し始めてから担当するセンシング領域へ移動し終えるまでの消費時間およびその際の全ノードの消費電力の合計値とする。また，消費時間および消費電力は，全試行の平均を結果として用いる。この評価指標を用いる理由は，想定環境でも述べたように本研究ではスタート地点からセンシング領域へ全ノードが効率良くたどり着くことがシステム全体のコスト低減につながり，また提案手法がセンシング時間や通信方式に非依存であるためである。

ノードは戦車型，車型の 2 種類を使用し，温度，湿度，赤外線の種類 3 種類のセンサを搭載してセンシングを行う。3 種類のセンサのセンシング範囲は実際のセンサの特性を考慮して図 7 のように定義した。

フィールドマップの広さ変化による消費電力量および消費時間の比較

フィールドマップの広さを 12m 四方から 20m 四方まで変化させ，全ノードの消費電力および消費時間について比較した。フィールドマップ上に以下に示す 2 通りの地形コストを配置し，経路探索を行った。

- 正方形の地形コストをランダムに配置（図 8(a)）
- 迷路（図 8(b)）

ノードは，戦車型および車型をそれぞれ 6 台ずつ，計 12 台利用し，表 1 に示すように各

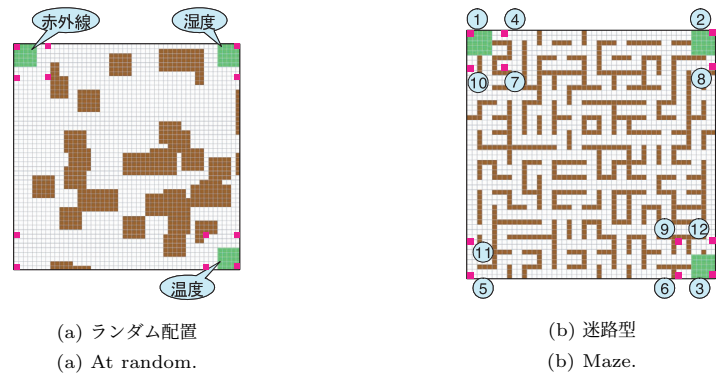


図 8 地形コストの例
Fig. 8 Example of geographical features.

表 1 使用するノード
Table 1 Assumed nodes.

ノード番号	種類	搭載するセンサ	ノード番号	種類	搭載するセンサ
1	戦車型	温度, 湿度	7	車型	温度, 湿度
2	戦車型	温度, 赤外線	8	車型	温度, 赤外線
3	戦車型	湿度, 赤外線	9	車型	湿度, 赤外線
4	戦車型	温度	10	車型	温度
5	戦車型	湿度	11	車型	湿度
6	戦車型	赤外線	12	車型	赤外線

ノードには 2 種類または 1 種類のセンサを搭載する。ノード ID が小さいほど優先度が高いものとし、優先度が低いノードは優先度が高いノードと衝突を回避するような経路が探索される。図 8 (a) に示すようにセンシング領域はフィールドマップの隅 3 カ所とし、温度、湿度、赤外線の 3 種類のセンサによるセンシングが要求されている。以下に、フィールドマップの広さが 100 セル四方の場合における問合せ構文の例を示す。

```
SELECT temp, 1
FROM (96, 96), (4, 4)
SELECT humid, 1
FROM (96, 0), (4, 4)
SELECT infrared, 1
```

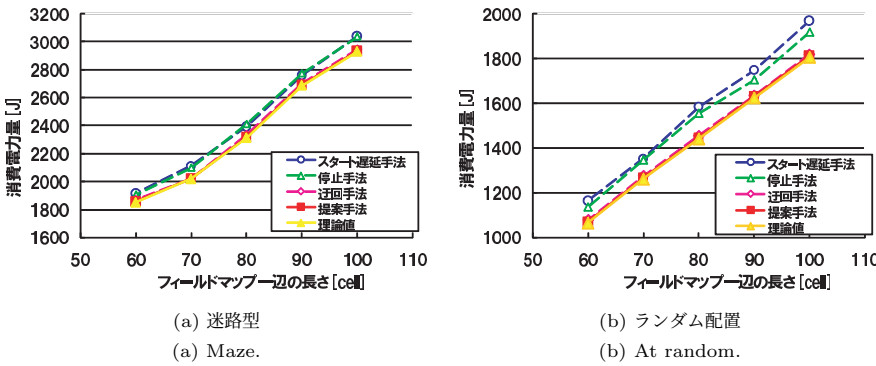


図 9 フィールドマップの広さに対する消費電力量
Fig. 9 Power consumption vs. size of field maps.

FROM (0, 0), (4, 4)

図 9 (a), (b) はそれぞれ迷路状、ランダムな地形コスト配置を想定した経路探索により得られた全ノードの消費電力の合計を示す。ここで理論値は、同一セルに複数ノードが重なってもノードどうしが衝突しないと仮定し、従来アルゴリズムにより経路を求めた結果である。提案手法は、従来のアルゴリズムにより得られた結果を単純に足し合わせた理論値と比較して若干余分に電力を消費するが、地形コストの配置およびフィールドマップの広さにかかわらず、スタート遅延手法と比較して平均 5.9%、停止手法と比較して平均 4.6%消費電力を低く抑えている。比較手法では、いったんノードが動き始めると停止することなくゴール地点に到達できるようスタート時間を遅らせる、または衝突を検出した直前のセルで停止して衝突を回避するため、停止、待機および動き始める動作に余分な消費電力を要する。一方、提案手法では他ノードとの衝突を回避すると同時になるべく最小コストとなる経路を探索するため、結果的に 2 種類の比較手法よりも消費電力が低く抑えられている。

一方、他の経路を探索することにより衝突を回避する手法と比較して、提案手法は消費電力量が 0.4%低く抑えられているが、ほぼ漸近している。これは、提案手法では他のノードとの衝突を回避するために、主に他の経路が選択されたためである。

また、地形コストをランダムに配置した場合、迷路状に配置した場合の結果と比較して提案手法がより有効である。たとえば、提案手法による消費電力量はランダムに地形コストを配置した場合、スタート遅延手法と比較して 8.4%、停止手法と比較して 6.1%それぞれ

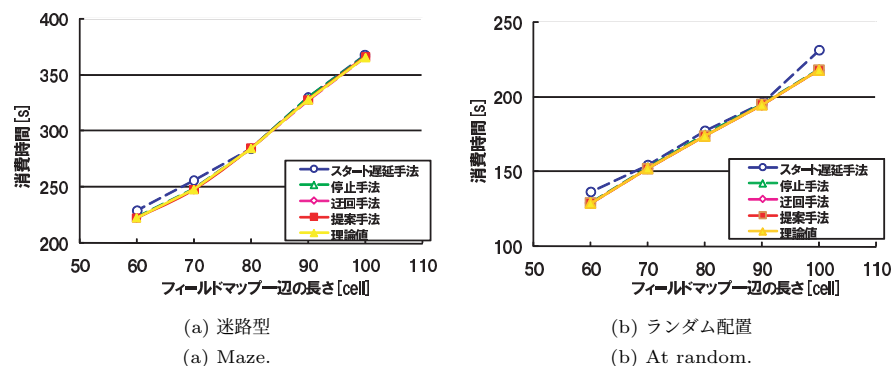


図 10 フィールドマップの広さに対する消費時間
Fig. 10 Time consumption vs. size of field maps.

低減されているのに対し、迷路状に配置した場合では、スタート遅延手法と比較して 3.4%、停止手法と比較して 3.2%の低減にとどまっている。迷路状に地形コストが配置された環境ではノードが移動するための経路が複数通り存在し、従来手法により得られた結果からノードどうしが衝突する回数がより少ないのに対し、地形コストがランダムに配置された環境では、最小コストとなる経路が複数ノードで重複する可能性が高く、他の経路を探索することで衝突を回避できる提案手法は比較手法よりも消費電力を低く抑えられる。

図 10 (a), (b) はそれぞれ迷路状、ランダムな地形コスト配置を想定した経路探索による消費時間の結果を示す。地形コストの配置、フィールドマップの広さにかかわらず、スタート遅延手法と比較して平均 2.3%、停止手法と比較して平均 0.4%消費時間が短縮されているが、消費電力量ほど改善されていない。上述したように消費時間とはすべてのノードが配置し終わるまでの時間を示しており、移動距離が最も長いノードの移動時間に集約されている。そのため、最も移動時間を要するノードが他のノードとの衝突を回避するために停止しない限り、消費時間には影響しない。

5. 考 察

5.1 ノードの巡回について

現在は、ノードはスタート地点からセンシング領域へ移動した後は継続的にセンシングし続け、他のセンシング領域へは移動しないものとしている。しかし、移動型ノードを用いた

センサネットワークでは、ノードは複数個所のセンシング領域を巡回して継続的にセンシングを行うことがある。そこで、現在は 1 つのノードにおいてスタート地点から 1 カ所のセンシング領域への移動経路を求めているが、2 カ所目、3 カ所目というように複数個所のセンシング領域を循環する順番をすべて列挙し、それぞれについて提案するアルゴリズムにより経路探索を行ったうえで、他のノードの計算結果との組み合わせの中から、消費電力の合計が最小となる経路を決定することを考えている。

5.2 考慮できていない実用上の問題

本稿で提案する方式を用いることで、これまでに実用上問題であったノードどうしの衝突を回避した経路探索が可能となった。一方、下記のように未解決の問題も多く残っている。

現在ベースコストは、周辺のセルへ手でノードを移動させることで測定している。そこで、ユーザの負担を減らすべく自動的に測定できるプログラムの作成を検討している。また、現在経路探索に使用しているベースコストにおけるセルの大きさは、ノードごとに最適な大きさを決定している。今後さらに様々なノードの利用に対応するべくベースコストのセルの大きさをさらに小さく区切り、大きさの異なるノードを統一的に扱う手法を検討する。

実機は、本稿で提案したアルゴリズムにより得られた経路に沿って移動する際に、タイヤのスリップや地形の細かい変化、さらには使用するノード自体の動作精度の低さにより、指定された経路と実際にノードが通過する経路に誤差が生じる。そのため、ノードの位置を推定し、移動誤差が生じていれば指定された経路上へノードを移動させなければならない。ノードの位置推定の手法としては、加速度センサやジャイロセンサといった内界センサを利用するデッドレコニングと、カメラなどの外界センサを用いる手法に大別できる。外界センサは一般的に高価であり、センサの分解能やノイズなどにより誤差が accrue する恐れがある。一方、内界センサはノードに複数搭載することでより高精度の位置推定が可能となるが、ノードが大型になったり高価になったりする。

ノードの位置推定に関しては、複数の光学マウスセンサを利用して移動距離を調節床面から読み取ることで位置推定を行う研究⁶⁾ がなされており、車輪の変形やスリップの影響をなるべく受けない高精度のデッドレコニングを実現している。そこで、本研究においてもノードに安価な内界センサを搭載することを考えている。

5.3 問合せ言語の検討

本稿で提案した問合せ言語では、移動型ノードを利用したセンサネットワークで使用するための従来の問合せ言語からの改良点としてセンシング精度の要素を新たに設けている。一方、TinyDB⁴⁾ などの研究で実現できている記述で提案手法では利用できない記述も多い。

移動型センサネットワークのための問合せ言語を確立することは、本研究において重要な要素の1つであり、今後柔軟で記述能力の高い問合せ言語を提案する予定である。また、本研究では今後ノードがセンシング領域へ移動後さらに異なるセンシング領域へ移動することを想定しており、このような要求を記述できるように問合せ言語を改良することも必要となる。具体的には、センサデータ取得値に基づく再帰設定など、別の問合せ結果をもとに、次に要求するセンサデータを決定するような仕組みが必要となる。

また、現在は、1台のノードが1カ所のセンシングを行うことを想定しているが、センシング精度を上げるための手段としては複数ノードの同時配置も考えられる。そこで、現在のセンシング精度をさらに細かく設定できるようにし、センシングを行う台数などの条件を指定できるようにすることを考えている。

5.4 通信範囲の問題

移動型ノードを用いたセンシングにおいて、現在はノードがセンシング領域へ移動する段階に着目して研究を進めており、センシングを行った後にデータセンタにセンサデータを転送することを考慮していない。実際には、センサデータに制限時間が設定されるとデータセンタまでデータを制限時間内に到達させなければならないため、ルーティングプロトコルについても検討しなければならない。しかしこの問題に対しては、センシング領域へ移動した後データセンタまで戻ってくる経路を探索することで解決できると考えている。もし、計算結果から、ノードがセンシング後にデータセンタへ帰ってくると指定された時間に間に合わない場合でも、センサデータを転送することを念頭において、あらかじめ必要なノードを転送可能な位置に配置させることで対処できる。

5.5 関連研究

これまでの移動型センサネットワークに関する研究としては、複数のノードが協調して様々な状況に応じたルーティングを行うことを目的としたRAMOS⁸⁾や、ノードの正確な位置特定を目的とした研究¹⁰⁾、さらにセンシングカバレッジの拡大を目的とした研究⁹⁾が行われている。しかしこれらの研究では、本研究で提起した実用上の問題が考慮されていない。

また、ノードが人や動物に寄生して移動するという新しい移動手法を提案し実際に実機の製作を行っている例として、Parasitic Mobility³⁾があげられる。しかしParasitic Mobilityは寄生対象がつねに存在する環境を想定しており、人や動物が近づけない危険地帯へのノードの配置は不可能であるため、移動型センサネットワークの特徴を活かしきれていない。

今枝らは、センサノードに関する専門的な知識がないユーザでもアプリケーションを容易

に利用できるようにする、ROASEN システムを提案している²⁾。アプリケーション開発者はセンサノードの配置条件をXML形式で記述することで、センシング目的に応じたノード配置が可能となる。しかし、XMLではノードの配置場所について、オブジェクトや壁からの距離など詳細な条件を設定する必要があり、ユーザの負担となる。また、アプリケーションごとにXMLを記述しなければならない、あまり汎用的ではない。本稿で提案する手法では、問合せ言語を用いてセンシング領域やセンサの種類などを指定するだけで、ノードの配置場所なども自動的に決定できる。

6. 結 論

本稿では、筆者らがこれまでに提案してきたコストマップを用いた経路探索アルゴリズムでは考慮できなかったノードどうしの衝突の問題を解決する、複数ノードの衝突を考慮した経路探索アルゴリズムを提案した。これにより、衝突を効率的に回避する経路探索が可能となった。さらに、移動型ノードを用いたセンサネットワークのための問合せ言語を提案した。ユーザは、センシングしたい場所、センサの種類などの条件を容易に設定でき、複数ノードをセンシングの目的に応じて効率かつ自動的に配置できるようになった。本研究では、スタート時間を遅らせることで衝突を回避するアルゴリズムと提案手法をシミュレーションにより比較し、提案手法の有効性を示した。

今後の課題として、ノードがセンシング領域へ移動後さらに別のセンシング領域へ移動する状況への対応、それにともない増加する計算量の削減、より柔軟な問合せ言語への拡張があげられる。さらに、得られたセンサデータの転送、ノードの移動誤差への対応、消費電力量の分散を低減させる手法についても検討する予定である。

謝辞 本研究の一部は、文部科学省基盤研究(A)(17200006)の研究助成によるものである。記して謝意を表す。

参 考 文 献

- 1) Bourg, D.M. and Seemann, G.: *AI for Game Developers*, Oreilly (2004).
- 2) 今枝卓也, 大澤 亮, 高汐一紀, 徳田秀幸: アプリケーション適応型センサノード配置ロボットの提案, 情報処理学会研究報告(ユビキタスコンピューティングシステム研究会 2006-UBI-11), Vol.2006, No.119, pp.29-35 (2006).
- 3) Laibowitz, M. and Paradiso, J.A.: Parasitic mobility for pervasive sensor networks, *Proc. 3rd International Conf. on Pervasive Computing (PERVASIVE 2005)*, pp.255-278 (2005).

- 4) Madden, S., Franklin, M.J., Hellerstein, J.M. and Hong, W.: TinyDB: An Acquisitional Query Processing System for Sensor Networks, *ACM Trans. Database Systems (TODS)*, Vol.30, Issue 1, pp.122–173 (2005).
- 5) 中宮正樹, 岸野泰恵, 寺田 努, 西尾章治郎: コストマップを用いた移動型センサノードの経路探索手法, *情報処理学会論文誌*, Vol.49, No.3 (2008).
- 6) 関森大輔, 宮崎文夫: 複数の光学マウスセンサを用いた移動ロボットのデッドレコニング, *計測自動制御学会論文集*, Vol.41, No.10, pp.775–782 (2005).
- 7) Sibley, G.T., Rahimi, M.H. and Sukhatme, G.S.: Robomote: A tiny mobile robot platform for large-scale ad-hoc sensor networks, *Proc. IEEE International Conf. on Robotics and Automation (ICRA 2002)*, pp.1143–1148 (2002).
- 8) Suzuki, R., Makimura, K., Saito, H. and Tobe, Y.: Prototype of a sensor network with moving nodes, *Proc. 1st International Workshop on Networked Sensing Systems* (2004).
- 9) Wang, G., Cao, G. and Porta, T.L.: Movement-assisted sensor deployment, *Proc. Conf. on Computer and Communications Societies (INFOCOM 2004)*, Vol.23, No.1, pp.2469–2479 (2004).
- 10) Tilak, S., Kolar, V., Abu-Ghazaleh, N.B. and Kang, K.D.: Localization for mobile sensor networks, *Proc. International Conf. on Mobile Computing and Networking*, pp.45–57 (2004).
- 11) Crossbow. <http://www.xbow.jp/>
- 12) MindStorms. <http://mindstorms.lego.com/>

(平成 20 年 7 月 7 日受付)

(平成 21 年 1 月 7 日採録)



中宮 正樹

2006 年大阪大学工学部電子情報エネルギー工学科情報システム工学科目卒業。2008 年同大学院情報科学研究科マルチメディア工学専攻博士前期課程修了。同年株式会社 NTT データ入社、現在に至る。大阪大学大学院では移動型センサネットワークの研究に従事。



寺田 努 (正会員)

1997 年大阪大学工学部情報システム工学科卒業。1999 年同大学院工学研究科博士前期課程修了。2000 年同大学院工学研究科博士後期課程退学。同年より大阪大学サイバーメディアセンター助手。2005 年より同講師。2007 年神戸大学大学院工学研究科准教授。現在に至る。2004 年より特定非営利活動法人ウェアラブルコンピュータ研究開発機構理事、2005 年には同機構事務局長を兼務。2004 年には英国ランカスター大学客員研究員を兼務。博士 (工学)。アクティブデータベース、ウェアラブルコンピューティング、ユビキタスコンピューティングの研究に従事。IEEE、電子情報通信学会、日本データベース学会、ヒューマンインタフェース学会の各会員。



西尾章治郎 (正会員)

1975 年京都大学工学部数理工学科卒業。1980 年同大学院工学研究科博士後期課程修了。工学博士。京都大学工学部助手、大阪大学基礎工学部および情報処理教育センター助教授、大阪大学大学院工学研究科情報システム工学専攻教授を経て、2002 年より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教授となり、現在に至る。2000 年より大阪大学サイバーメディアセンター長、2003 年より大阪大学大学院情報科学研究科長、その後 2007 年より大阪大学理事・副学長に就任。この間、カナダ・ウォータールー大学、ビクトリア大学客員。データベース、マルチメディアシステムの研究に従事。現在、Data & Knowledge Engineering 等の論文誌編集委員。本会理事を歴任。本会論文賞を受賞。電子情報通信学会フェローを含め、ACM、IEEE 等 8 学会の各会員。