

推薦論文

無線端末の遭遇履歴情報を用いた 移動軌跡推定手法の提案

藤井 彩恵^{†1} 内山 彰^{†1} 梅津 高朗^{†1,†2}
山口 弘純^{†1,†2} 東野 輝夫^{†1,†2}

本論文では、正確な位置情報を発信する固定ノード(ランドマーク)や他の移動ノードとの遭遇情報を収集し、それらを用いて移動ノードの軌跡をオフライン(非リアルタイム)で推定する手法を提案する。提案手法では、ランドマーク間を最も直線に近い軌跡で移動したと考えられるノードの移動軌跡を推定し、その移動軌跡を他のノードの軌跡の推定に用いるという処理を繰り返す。さらに、シミュレーテッド・アニーリング(SA)を用いて、全移動端末の軌跡を一括して修正することにより、移動軌跡の精度を向上させる。シミュレーション結果より現実的な環境下で推定誤差が最大無線到達距離の40%程度に抑えられることを確認した。

Estimating Movement of Wireless Terminals Using Ad hoc Communication History

SAE FUJII,^{†1} AKIRA UCHIYAMA,^{†1} TAKA AKI UMEDU,^{†1,†2}
HIROZUMI YAMAGUCHI^{†1,†2} and TERUO HIGASHINO^{†1,†2}

In this paper, we design and implement an algorithm to estimate the movement of wireless terminals. The proposed method relies on the history of ad hoc wireless communication between those terminals and the landmark stations to track the movement of each terminal. The principle of the algorithm design lies in iterative refinement of their positions so that they finally settle in appropriate positions that satisfy the constraints derived from the given communication history. We have evaluated the performance of our algorithm by simulations and confirmed that the average position estimation error was less than 40% of the wireless range with realistic settings.

1. はじめに

近年、モバイル情報端末やセンサによる位置測位情報を利用した位置情報サービスが注目されている。その一方で、そのような測位情報を収集して得られる移動軌跡情報の様々な応用に関する期待も高まっている。現在、携帯端末や車載端末ではGPSが普及しているため、屋外においては端末の移動軌跡をリアルタイムで比較的容易に取得できるが、駅や地下街など衛星からの電波が届かない領域では不向きである。これに対し、たとえば文献1)のように端末のアドホック無線通信機能を用いて位置推定を行う方法も多く提案されている。既存手法の多くは、位置基準局からの情報伝達と距離の推定をマルチホップアドホックネットワークを利用して行うため、通信切断が頻繁に起こることが予想されるモバイルアドホックネットワークにおいては現実的ではない。

本論文では、位置基準局(以下、ランドマーク)と移動無線端末(以下、移動ノード)が蓄積した他ノードとの遭遇情報(いつ、どのノードと遭遇したのかの記録)のみを利用する低コストな移動軌跡推定手法TRACKIEを提案する。提案手法では、ランドマーク間を直線に近い軌跡で移動したと考えられるノードの移動軌跡を推定し、その移動軌跡を他のノードの軌跡の推定に用いるという処理を繰り返す。さらに、シミュレーテッド・アニーリング(SA)を用いて、全移動端末の軌跡を一括して修正することにより、移動軌跡の精度を向上させる。提案手法をネットワークシミュレータMobiREAL^{2),3)}を用いて評価した結果、 R_{max} を最大無線到達距離としたとき、平均位置誤差が約 $0.4R_{max}$ 程度であることを確認した。

2. 関連研究

既存の位置推定手法は、超音波や指向性アンテナなどを利用して電波到達距離や角度を測定するレンジベース手法^{4),5)}と、本論文で採用するような各ノード間の無線の到達情報のみを利用するレンジフリー手法^{1),6)–9)}とに分類できる。レンジフリーには、Centroid⁶⁾およびMCL⁷⁾のように各ノードがランドマークから直接受信した位置情報を用いる手法や、

†1 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University, Japan

†2 独立行政法人科学技術振興機構, CREST
Japan Science and Technology Agency, CREST

本論文の内容は2007年7月のマルチメディア, 分散, 協調とモバイル(DICOMO2007)シンポジウムにて報告され, ITS研究会主査により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である。

Amorphous¹⁾のようにランドマークからマルチホップで受信した位置情報を用いる手法、UPL⁸⁾のように他ノードの推定結果を逐次的に利用する手法がある。また、MDS-MAP⁹⁾のようにノード間の接続情報から各ノードの相対位置を計算し、ランドマークの位置座標から全ノードの絶対座標を決定する手法も提案されている。

一方、センサネットワークの分野では移動ノードの位置追跡手法が提案されている¹⁰⁾⁻¹²⁾。これらの手法では、人感センサなどを備えたノードを高密度に配置し、各ノードは検知したイベントを基地局へと送信する。基地局では収集されたイベント情報を統合することにより、対象ノードの位置追跡を行う。これらの位置追跡手法は多数の固定センサノードをランドマークとして利用する点で TRACKIE とは実行環境やアプローチが大きく異なる。

3. TRACKIE の概要

3.1 想定環境

提案手法では、移動ノードと呼ばれる移動無線端末群と、ランドマークと呼ばれる位置基準ノード群の存在を仮定し、これらをあわせてノードと呼ぶ。移動ノードは、ランドマークが疎に配置された領域内を移動するものとする。ノードはそれぞれ個別の ID を与えられ、ID はノードの種別（移動ノードまたはランドマーク）も表すものとする。すべてのノードは ZigBee や Bluetooth などの近距離無線通信デバイスにより、互いに通信可能であるとす、最大通信距離は同じであるとする（これを R で表す）。各ノードは定期的に自身の ID を Hello パケットとしてブロードキャストする。隣接ノード j から Hello パケットを受信したノード i は、Hello パケットを受信した時刻 t 、自身の ID_i および受信した Hello パケットに含まれる ID_j の組 (t, ID_i, ID_j) を端末に蓄積する。以降、この組を遭遇エントリと呼ぶ。なお、すべてのノードが保持する時計はおおよそ同期しているものと仮定する。移動ノードはランドマークと遭遇したとき、端末に記録された全遭遇エントリをランドマークを介してサーバへ送信する。

提案手法では、サーバに収集された遭遇エントリとランドマークの位置情報に基づき N 個の移動ノードの移動軌跡を推定する。ここで、一般性を失うことなく、遭遇エントリに含まれる時刻 t は $[0, T]$ に含まれる整数であるとする（ T はある自然数）。また、移動軌跡は時刻ごとの通過点（footprint）の列から構成されるものとし、ノード i の時刻 t における通過点を $p_{i,t}$ と定義する。すなわち、ノード i の移動軌跡は $p_{i,0}, p_{i,1}, \dots, p_{i,T}$ で表される。よって、 $i \in N$ と $t \in [0..T]$ の組ごとに $p_{i,t}$ の位置座標を出来る限り正確に決定することを目標とする。

3.2 アルゴリズムの基本アイデア

ノードの移動軌跡を高精度に推定するための方法として多くのランドマークを配置することが考えられるが、一般に多数のランドマークを配置することは、設置場所の確保や設定の手間などのコストが無視できない。一方で、移動軌跡導出の対象者に簡易で安価な近距離無線端末を保持させる代わりにランドマーク数を削減できれば、設置などにかかるコストを抑制した移動軌跡推定法として有用であると考えられる。このため、多くのランドマークを仮定するのではなく、移動ノードが適当な時間間隔でランドマークに遭遇することを期待し、それぞれのランドマーク間を移動した軌跡（以下、部分軌跡と呼ぶ）を正確に求めることを目標とする。以下では、すべてのノードは 2 回以上ランドマークに遭遇し、その移動軌跡は 1 つ以上の部分軌跡から構成されるものとする。部分軌跡を推定するための最も単純な方法として、各遭遇エントリに含まれる 2 つのノードの通過点の距離が最大通信距離 R 以下であるといった制約を与え、線形計画法などを用いて各通過点の位置座標を決定する方法などが考えられるが、移動ノード数の増加にとともに、制約の数が莫大となるため、このような手法は現実的ではない。

TRACKIE では、すべてのノードの部分軌跡のうち比較的高精度な推定が可能であると考えられる部分軌跡をはじめに検出する。次に、その部分軌跡を構成する通過点を新たなランドマーク（擬似ランドマーク）として扱うことにより、他の部分軌跡を決定するプロセスを繰り返す。ここで最初に選ぶ部分軌跡（最初に擬似ランドマークとなる部分軌跡）は移動ノードがランドマーク間を等速直線移動をしたと仮定する。一般に、歩行者は必ずしも等速直線的に歩き続けるとは限らない。しかし、歩行者の数が多い場合、その中の一部の歩行者はランドマーク間を等速直線状に移動していると期待できる。

このようなアイデアに基づき、提案手法では 3 つのフェーズから構成されるアルゴリズムを考える。1 番目を初期化フェーズと呼び、ランドマークとの遭遇エントリのみを用いて移動軌跡の初期解を与える。ここでは、各移動ノードは 2 つのランドマーク間を等速直線移動をしたと仮定し、初期解を与える。ここで、実際の移動軌跡を実軌跡、推定された移動軌跡を推定軌跡と呼ぶ。また、初期化フェーズで得られる推定軌跡を初期軌跡と呼ぶ。たとえば、図 1 (a) に示す実軌跡の場合、その初期軌跡は図 1 (b) のようにランドマークの位置座標を頂点とするような線分から構成される。

2 番目は逐次修正フェーズと呼び、移動ノード間の遭遇エントリを用いて初期軌跡の形状を修正する。まず、時刻 t にノード i とノード j が遭遇したことを示す遭遇エントリから、時刻 t においてノード i とノード j の通過点の距離は最大通信距離 R 以内でなければなら

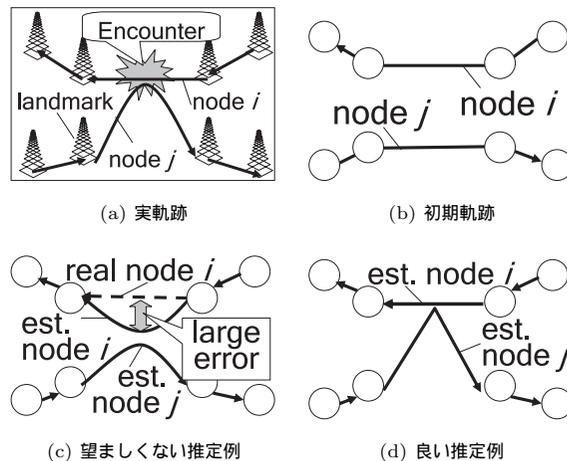


図 1 TRACKIE による移動軌跡の推定

Fig. 1 Estimation of trajectories by TRACKIE.

ないという制約が考えられる．このような制約を遭遇制約と呼ぶ．遭遇制約を満たすように軌跡を修正するには，2 ノードの部分軌跡が互いに近づくように曲げるという方法が考えられる．しかし，もし図 1 (a) のように一方の実軌跡が直線状であった場合，図 1 (c) のように曲げると推定誤差が大きくなる．このため，逐次修正フェーズでは，初期化フェーズで得られた等速直線状の部分軌跡群のうち，実軌跡に最も近いと考えられる部分軌跡をはじめに決定し，他の部分軌跡をそのような部分軌跡へ近づけるように修正する．たとえば，図 1 (d) のように，ノード i の実軌跡が等速直線状であると決定できれば，ノード j の移動軌跡を遭遇制約によりノード i の移動軌跡に近づけることで，より現実に近い軌跡推定が行える．

最後のフェーズでは，逐次修正フェーズで得られた推定軌跡について，建物などの障害物内部を通過したり，移動速度や方向の変動が極端に大きいといった不自然な行動がないかを評価し，もし移動速度や方向の変動が大きい場合はより小さい推定軌跡へと修正する．ただし，あるノードの推定軌跡を修正する場合，それに制約を受ける他ノードの軌跡も合わせて修正する必要があるため，極端に大きく変更するような修正は望ましくない．したがって，すべての遭遇制約を満たしながら個々の軌跡における移動速度や方向の変動が小さい軌跡となるよう，軌跡全体を考慮しながら徐々に修正していく．このような修正にはすべての通過点について評価する必要があり，非常に多くの解候補が存在する．また，制約式の多くが位

置座標間の関係式で与えられる多元 2 次式である．よってヒューリスティックアルゴリズムの設計は非常に困難であり，シミュレーテッド・アニーリング (SA) や遺伝的アルゴリズム (GA) など探索系のアルゴリズムが最も望ましいと考えられる．今回は探索系のアルゴリズムの中で実装がシンプルに行えることから，SA に基づく最適化技術を用いて解空間の探索を行う．このフェーズを SA 修正フェーズと呼ぶ．

4. アルゴリズムの設計

初期化フェーズおよび逐次修正フェーズでは，通過点ごとに状態を設定する．通過点の状態は固定状態 (anchored)，擬固定状態 (quasi-anchored)，束縛状態 (constrained)，自由状態 (free) のいずれかであるとする．通過点 $p_{i,t}$ が固定状態となるのは，ノード i が時刻 t にランドマークと遭遇したとき，かつそのときに限るものとする．固定状態以外の状態については後述する．以下では，アルゴリズムの 3 つのフェーズについて説明する．

4.1 初期化フェーズ

はじめに，時刻 t にランドマークと遭遇したノード i の通過点すべてについて， $p_{i,t}$ の状態を固定状態に設定する．固定状態に設定された通過点の位置座標はランドマークの位置座標とし，以降のアルゴリズムを通して変更されない．また，ノード i の移動軌跡のうち，固定状態に設定された 2 つの通過点 $p_{i,t}, p_{i,t+l}$ ($l > 0$) 間のすべての通過点列を自由状態とし， $p_{i,t}$ と $p_{i,t+l}$ 間を等速直線移動をしたものとして，その位置を決定する．すなわち初期軌跡はランドマーク間を線分で結んだ形状となる．図 2 (a) にある実軌跡の例を示す．図 2 (b) に $p_{i,0}, p_{i,40}, p_{j,0}, p_{j,40}$ が固定状態に設定された初期軌跡を示す．

4.2 逐次修正フェーズ

逐次修正フェーズにおける初期軌跡の修正手順について述べる．通過点の状態として，擬固定状態を導入する．あるノードの通過点が擬固定状態になるとき，それ以降，その通過点の位置座標は固定されたものとして扱い，他の通過点の位置座標の推定に用いる．逐次修正フェーズでは，初期軌跡の中でより直線に近いと推定される部分軌跡から順に，その部分軌跡を構成する通過点を擬固定状態とし，固定状態や擬固定状態に設定された通過点との遭遇制約から，他の通過点の位置を修正する．すなわち実軌跡との誤差が大きいと推定される部分軌跡ほど修正の機会が多く与えられる．この処理は，非固定状態 (固定状態または擬固定状態でない状態) であるすべての通過点が擬固定状態に設定されるまで繰り返し行われる．ここで，擬固定状態とする通過点の候補決定方法と，新たにいくつかの通過点が擬固定状態とされた場合の非固定状態の通過点位置修正方法をそれぞれ説明する．

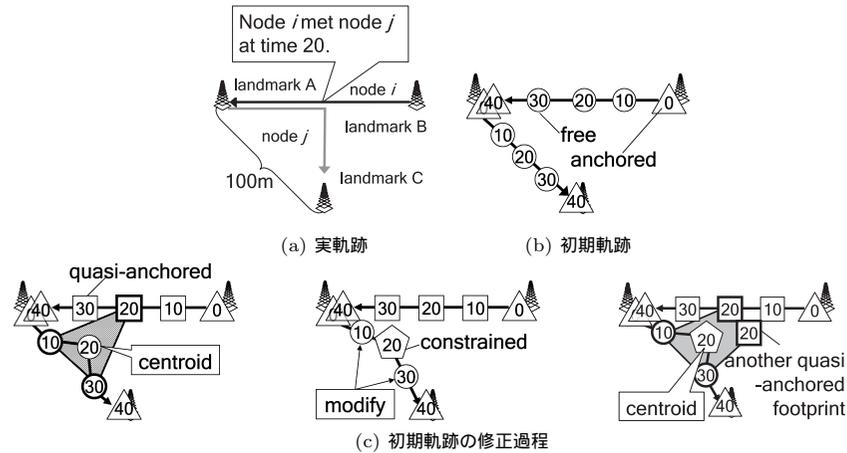


図 2 Exemplification of iterative modification phase.

図 3 に逐次修正フェーズのアルゴリズム記述を与える。

まず、疑似固定状態とする通過点の候補決定方法を説明する。 $\vec{v}_{i,t}$ を初期軌跡における時刻 t のノード i の推定速度ベクトルとする。 $q_{i,t}$ を初期軌跡における通過点 $p_{i,t}$ の位置座標とすると、速度ベクトル $\vec{v}_{i,t}$ は $\frac{q_{i,t} - q_{i,t+1}}{\Delta t}$ で定義される。初期化フェーズ後、 $i \in N$ と $t \in [0..T]$ のすべての組について速度ベクトルの大きさ $|\vec{v}_{i,t}|$ が計算される。疑似固定状態とする通過点の候補決定方法は、非固定状態である部分軌跡から、その実軌跡がランドマーク間を最も直線に近い形状で結ぶと思われる部分軌跡を選択する。ノードの移動速度に大きな差がない場合、これはランドマーク間を最も速く移動したノードの部分軌跡を選択することに相当する。よって、計算された速度ベクトルを用いて (1) $p_{i,t}$ と $p_{i,t+l}$ が固定状態である (2) $p_{i,t+1}, \dots, p_{i,t+l-1}$ がいずれも非固定状態である (3) $p_{i,t+1}, \dots, p_{i,t+l-1}$ での移動速度 (初期軌跡の性質上、同じ部分軌跡を構成する通過点ごとの移動速度の大きさ $|\vec{v}_{i,t+1}|, \dots, |\vec{v}_{i,t+l-1}|$ はすべて等しい) が最も大きい、の 3 条件を満たすような部分軌跡 $p_{i,t+1}, \dots, p_{i,t+l-1}$ を新たな疑似固定状態として選択する。

続いて非固定状態の通過点位置修正方法を説明する。ここで、通過点の状態として束縛状態を導入する。ある通過点は、非固定状態であり、かつ固定状態または疑似固定状態である通過点との遭遇制約に基づき位置が修正された場合のみ束縛状態となる。修正方法は、ま

```

Iterative Modification Phase
1 initialize NQ as set of all the free footprints
2 do{
3 //set some footprints to quasi-anchored
4 cand[] ← fastest_footprints(NQ)
5 for (each  $p_{i,t} \in \text{cand}[]$ ) {
6  $p_{i,t}.\text{state} \leftarrow$  "quasi-anchored"
7 NQ ← NQ -  $\{p_{i,t}\}$ 
8 }
9 //localize non-quasi-anchored footprints
10 for (each  $p_{i,t} \in \text{NQ}$ )
11 for (each  $p_{j,t} \in \text{encounter}(p_{i,t})$ ) {
12 if ( $p_{j,t}.\text{state} =$  "quasi-anchored")
13 EQ ← EQ  $\cup \{p_{j,t}\}$ 
14  $p_{i,t}.\text{pos} = \text{centroid}(p_{i,t-1}, p_{i,t+1}, \text{EQ})$ 
15 }
16 for (each constrained footprint  $p_{i,t}$ )
17 modifyFree( $p_{i,t}$ , nextConstrained( $p_{i,t}$ ))
18 }while (NQ  $\neq \phi$ )
    
```

図 3 逐次修正フェーズのアルゴリズム記述
Fig. 3 Algorithm description (Iterative modification phase).

ず固定状態や疑似固定状態である通過点と遭遇制約を持つような自由状態または束縛状態である通過点 $p_{i,t}$ を選択する。時刻 t にノード i と遭遇したノードのうち、ノード i との遭遇制約を満たさないような疑似固定状態または固定状態である通過点が存在する場合 (すなわち、 $p_{i,t}$ との距離が最大通信距離 R 以上である場合)、 $p_{i,t}$ を修正候補とし、その位置座標をその前後の通過点 $p_{i,t-1}$ 、 $p_{i,t+1}$ および遭遇制約を満たさない疑似固定状態または固定状態である通過点の重心にする。前後の通過点を考慮するのは、1 回の推定での位置の大きな変動を避けるためである。ここで、 $p_{i,t}$ の状態が自由状態の場合はその状態を束縛状態に変更する。次に、疑似固定状態または固定状態である通過点と遭遇制約を持たない通過点に関しては、自由状態ではない通過点間の通過点列が直線状となるように位置を修正する。すなわち、自由状態ではない通過点 $p_{i,t-k}$ および $p_{i,t+k'}$ を用いて $p_{i,t-k+1}, \dots, p_{i,t-1}$ と $p_{i,t+1}, \dots, p_{i,t+k'-1}$ の位置座標を $p_{i,t-k}$ 、 $p_{i,t}$ 間および $p_{i,t}$ 、 $p_{i,t+k'}$ 間が直線となるようそれぞれ修正する。

図 2 (c) に逐次修正フェーズによる初期軌跡の修正過程を示す。はじめに、通過点 $p_{i,10}$ 、 $p_{i,20}$ 、 $p_{i,30}$ が疑似固定状態に設定される。 $p_{i,20}$ が疑似固定状態となり、ノード i とノード

j が時刻 20 に遭遇しているため、 $p_{j,20}$ の位置を $p_{j,10}$, $p_{j,30}$ および $p_{i,20}$ の重心とし、それを束縛状態とする。次に、 $p_{j,0}$, $p_{j,20}$ 間および $p_{j,20}$, $p_{j,40}$ 間が直線となるよう、 $p_{j,10}$ と $p_{j,30}$ の位置を修正する。最後に、通過点 $p_{j,10}$, $p_{j,20}$, $p_{j,30}$ が擬似固定状態に設定され、アルゴリズムが終了する。

4.3 SA 修正フェーズ

シミュレーテッド・アニーリング (SA) は大局的な最適解の探索に広く用いられる手法である。SA では、現在の解から近傍解を作成し、それぞれの解をある評価関数で評価する。そして、現在の解と新しい解の評価値の差 $\Delta Cost$ から受理関数 $P(\text{rand}(0,1) < e^{-\Delta Cost/T})$ に従い、新しい解を採択するかを判定する処理を繰り返し行う。もし $\Delta Cost \leq 0$ であれば、 $e^{-\Delta Cost/T}$ は 1 より大きいいため新しい解はつねに採択される。一方 $\Delta Cost > 0$ であれば、確率的に受理され、 $\Delta Cost$ が小さいほど受理される確率が大きくなる。このように評価値が悪化する場合でも受理されるため、局所解に陥ることなく最適解を探索できる。

SA 修正フェーズでは文献 13) で Corana らが提案した SA を用いて逐次修正フェーズで修正された推定軌跡をさらに修正する。図 4 に SA 修正フェーズのアルゴリズム記述を示す。新しい解の生成 (図 4 の Neigh 関数に相当) では、現在の解を構成する通過点をランダムに選択し、その位置を現在の位置座標を中心とする半径 r_{sa} の円に存在する任意の点へ移動する。 r_{sa} は解の受理率 p が 0.5 に近づくよう、式 (1) の関数を用いて定期的に修正される (図 4 の Adjustment 関数に相当する)。

$$r_{sa} = \begin{cases} r_{sa} \cdot \left(1 + c_u \frac{p-0.6}{0.4}\right), & p > 0.6 \\ \frac{r_{sa}}{\left(1 + c_u \frac{0.4-p}{0.4}\right)}, & p < 0.4 \\ r_{sa}, & \text{otherwise} \end{cases} \quad (1)$$

c_u は定数であり、文献 13) での検証結果から $c_u = 2$ が最良値である。よって、Corana らの提案する SA では一般に $c_u = 2$ が最良値であるとし、提案手法でも $c_u = 2$ とする。また、評価関数は下記に述べる 3 種類の関数、遭遇制約誤差評価関数、障害物制約誤差関数、速度および方向評価関数とし、それぞれ $Cost_a$, $Cost_b$, $Cost_c$ で表し、関数ごとに SA を順次実行する (それぞれ図 4 の Cost 関数に相当する)。このように複数の種類の制約を満たす解を発見する際、すべての制約を同時に扱う方法も考えられる。しかし、最適化の対象となる制約が増えれば増えるほど、解の評価に時間を要するため最適解への収束が遅くなる。ここでは 3 種類の制約を遭遇制約、障害物制約、速度制約の順に優先順位をつけ、はじ

SA-based Modification Phase

```

1 initialize Tr as set of all the footprints
2  $c \leftarrow Cost(Tr)$ ,  $count \leftarrow 0$ 
3  $Tr_{best} \leftarrow Tr$ ,  $c_{best} \leftarrow c$ ,  $r_{sa} \leftarrow InitVal$ 
4 for ( $t \leftarrow T_{init}$ ;  $t < T_{end}$ ;  $t \leftarrow t * K$ ) {
5   for ( $i \leftarrow 0$ ;  $i < I_{max}$ ;  $++i$ ) {
6     repeat  $Tr_{new} \leftarrow Neigh(Tr, r_{sa})$ 
7       until  $Constraint(Tr_{new})$ 
8      $c_{new} \leftarrow Cost(Tr_{new})$ 
9      $\Delta c = (c_{new} - c)/c$ 
10    if ( $\text{rand}(0,1) < e^{-\Delta c/t}$ ) {
11       $Tr \leftarrow Tr_{new}$ ,  $c \leftarrow c_{new}$ ,  $++count$ 
12      if ( $c < c_{best}$ ) {
13         $Tr_{best} \leftarrow Tr$ ,  $c_{best} \leftarrow c$ 
14      } } }
15  $Tr \leftarrow Tr_{best}$ ,  $c \leftarrow c_{best}$ ,  $p \leftarrow count/I_{max}$ 
16  $r_{sa} \leftarrow Adjustment(r_{sa}, p)$ 
17  $count \leftarrow 0$ 
18 }
```

図 4 SA 修正フェーズのアルゴリズム記述

Fig. 4 Algorithm description (SA-based modification phase).

めに $Cost_a$ を用いて遭遇制約からおおまかな軌跡を導出し、 $Cost_b$ および $Cost_c$ を用いてその近傍で障害物制約や速度制約をより多く満たす解を見つける。これにより、各制約を満たす解を見つける時間が短縮され、全体として最適解への収束に要する時間が短くなると考えられる。ただし、3 種類の制約ごとに SA を独立に順次実行することにより、たとえば障害物制約を満たすような解の探索時に、それまでに導出された解が満たしていた遭遇制約が満たされなくなる場合があってはならない。そこで、近傍解を評価する際には、あらかじめ現在の解が満たしている制約をすべて満たしているかを判定し、満たしている場合に限り評価の対象とする (図 4 の Constraint 関数に相当する)。

遭遇制約誤差評価関数 遭遇制約との矛盾が少ないほどより良い解であると考えられる。そこで、評価関数 $Cost_a$ を遭遇エントリごとに定義される遭遇制約誤差 $e_{i,j,t}$ の総和 $\sum_i \sum_j \sum_t e_{i,j,t}$ と定義する。なお、 $e_{i,j,t}$ は式 (2) で表す。 $Cost_a$ の値が小さいほど、遭遇制約との矛盾が小さく、より高精度な移動軌跡であるといえる。

$$e_{i,j,t} = \begin{cases} 0, & |\overrightarrow{p_{i,t} p_{j,t}}| \leq R \\ |\overrightarrow{p_{i,t} p_{j,t}}| - R, & \text{otherwise} \end{cases} \quad (2)$$

障害物制約誤差関数 障害物に関する情報が得られる場合、障害物を横切らないような移動軌跡を導出できると考えられる．そこで通過点 $p_{i,t}$ ごとに障害物制約誤差 $o_{i,t}$ を評価し、もし $p_{i,t}$ が障害物内部に存在する場合は $o_{i,t}$ を障害物の境界との最短距離として定義する． $Cost_b$ は障害物制約誤差の総和 $\sum_i \sum_t o_{i,t}$ として定義する． $Cost_b$ の値が小さいほど、障害物を横切らないような自然な移動軌跡であるといえる．ここで、障害物情報は建物などの障害物のおおよその位置が分かる簡易的なものを想定する．たとえば、Google Maps¹⁴⁾ などで提供されている地図情報を加工することにより得られるものとする．

速度および方向評価関数 一般にノードの速度や方向が突然変化するとはいえない．そこで速度や方向の変動がより小さい移動軌跡を導出することを考える．ここで \bar{V}_i をノード i の平均速度とし、 $\bar{V}_i = \sum_{t=0}^{T-1} |\vec{v}_{i,t}| / T$ として定義する．各時刻の速度と平均速度の差 $s_{i,t}$ は $|\vec{v}_{i,t}| - \bar{V}_i$ で定義される．また、前後の移動方向の差 $a_{i,t}$ を式 (3) で定義する．これより、ノードの速度や方向の変化量を $\sum_i \sum_t (s_{i,t} + a_{i,t})$ で表し、これを $Cost_c$ とする． $Cost_c$ の値が小さいほど、滑らかで変動の少ない移動軌跡であるといえる．

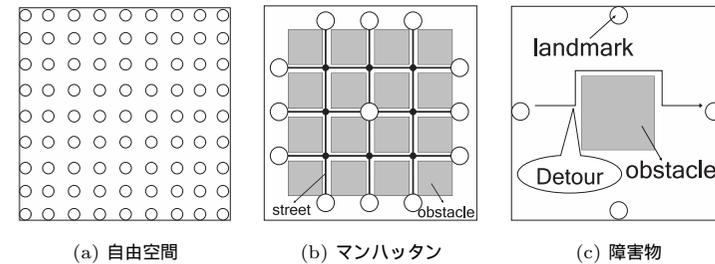
$$a_{i,t} = \frac{360}{2\pi} \times \arccos \left(\frac{\vec{v}_{i,t+1} \cdot \vec{v}_{i,t}}{|\vec{v}_{i,t+1}| |\vec{v}_{i,t}|} \right) \quad (3)$$

5. シミュレーション

TRACKIE の性能を評価するため、我々の研究グループが開発したネットワークシミュレータ MobiREAL^{2),3)} を用いてシミュレーションを行った．MobiREAL では 0.2 秒単位にノードの位置や移動方向の計算を行い、提案方式では 1 秒単位で計算された遭遇情報を用いるため、MobiREAL の計算精度が提案手法の検証結果に与える影響はないと考えられる．以下の実験では、ノードの無線通信方式は IEEE 802.11 とし、パケットの送信キューのサイズを 50,000 byte とする．

5.1 様々なシナリオにおける推定誤差の評価

様々な環境における異なる形状の推定軌跡の精度を評価するために、3 種類のシナリオを使用した．zigzag シナリオでは、各ノードは図 5 (a) に示すような 500 m 四方の自由空間マップ上で Random WayPoint (RWP) モビリティモデルに沿って移動するものとする．このシナリオにより折れ線上の形状から構成される移動軌跡の再現性を評価する．street walking シナリオでは、図 5 (b) に示すような幅 20 m の道路が 6 本配置された 500 m 四方の大きさのマンハッタンマップを用い、都市部を想定した移動軌跡の再現性を評価する．各ノードは道路に沿って移動し、各交差点で後方を除く方向をランダムに選択する、Random



(a) 自由空間 (b) マンハッタン (c) 障害物

図 5 シミュレーションのマップ (丸はランドマークを表す)
Fig. 5 Maps of simulated areas (circles represent landmarks).

Street Decision (RSD) モビリティモデルに従う．detour シナリオでは、図 5 (c) に示すような 100 m 四方の領域に 25 m 四方の大きさの障害物が中央に配置されたマップを利用し、各ノードはランドマーク間を障害物を回り込むように移動する．このシナリオにより、ランドマーク間を直線で移動する軌跡が存在しない場合における移動軌跡の再現性を評価する．なお、street walking シナリオとの違いは、ランドマーク間を直線移動するノードがまったく存在しない点である．

無線伝搬モデルは Line Of Sight (LOS) モデルとする．LOS モデルは、ノードが互に見通すことができ、かつノード間の距離が R_{max} 以下の場合に通信できるものとする．また TRACKIE で想定する最大通信距離 R は R_{max} と等しいものとする．移動ノードの速度は平均 2.0 m/s、分散 0.1 の正規分布に従うものとする．各シナリオにおけるノード数とランドマーク数はそれぞれ zigzag シナリオでは (3,000, 81)、street walking シナリオでは (1,000, 13)、detour シナリオでは (200, 4) とする．Hello パケットの送信間隔は 1 秒とする．

上記に述べたシミュレーション環境において 2 つの項目について評価した．1 つ目は実際の位置と推定位置との誤差を評価する指標として平均位置誤差を評価する．平均位置誤差は全時刻における各ノードの実際の位置と推定位置のユークリッド距離の平均値とし、式 (4) により定義される．ただし、 $p_{i,t}$ および $\hat{p}_{i,t}$ はそれぞれ実際の位置、推定位置とする．

$$\sum_i \sum_t |\overrightarrow{p_{i,t} \hat{p}_{i,t}}| / (T \cdot N) \quad (4)$$

2 つ目は実際の移動方向と推定された移動方向との誤差を評価する指標として平均角度誤差を評価する．平均角度誤差は時刻 t から時刻 $t+l$ (ここでは $l=10$) で定められる任意の時間

表 1 各シナリオにおける平均位置誤差および平均角度誤差の評価結果 (太字は各シナリオにおける最良値とする)
 Table 1 Average position errors / average angle errors (bold font indicates the best throughout scenario).

	zigzag (m) / (rad)		street walking (m) / (rad)	detour (m) / (rad)
	全領域	中央領域		
(i) 初期化フェーズ後	36.75 / 0.48	30.33 / 0.40	41.96 / 0.61	11.34 / 0.49
(ii) 逐次修正フェーズ後	5.42 / 0.22	3.97 / 0.17	3.77 / 0.13	7.69 / 0.45
(iii) SA 修正フェーズ後	5.42 / 0.22	3.97 / 0.17	3.71 / 0.12	4.76 / 0.38
(iii') SA 修正フェーズのみを適用した場合	26.93 / 0.42	19.28 / 0.32	19.29 / 0.42	6.52 / 0.48

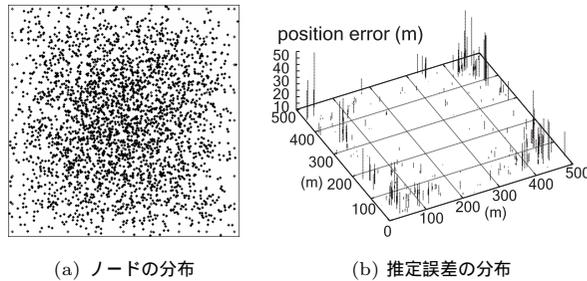


図 6 zigzag シナリオにおけるある瞬間における推定誤差
 Fig. 6 Positioning errors at a moment in zigzag scenario.

帯における実際の位置に基づく移動ベクトルと推定位置に基づく移動ベクトルの角度の差の平均値とし、式 (5) により定義される。ただし $avgv_{i,t} = (\overrightarrow{p_{i,t}p_{i,t+1}})/l$, $\widehat{avgv}_{i,t} = (\widehat{\overrightarrow{p_{i,t}p_{i,t+1}}})/l$ とする。

$$\sum_i \sum_t \arccos \left(\frac{avgv_{i,t} \cdot \widehat{avgv}_{i,t}}{|avgv_{i,t}| |\widehat{avgv}_{i,t}|} \right) / (T \cdot N) \quad (5)$$

表 1 に各シナリオの平均位置誤差と平均角度誤差の評価結果をアルゴリズムの 3 つのフェーズおよび SA 修正フェーズのみを適用した場合 (ケース (iii')) について掲載する。結果より, zigzag シナリオにおける全領域での平均位置誤差の値は 5 m 以上であり, 他の 2 つのシナリオよりも大きい。一方, 中央領域 (300 m 四方領域) では平均位置誤差の値は 4 m 以下であり, 他のシナリオでの結果に近い。これは, 図 6 (a) に示すように, RWP モデルではノードの分布が領域の中央に集中するという特性があり, 境界付近のノードは中央付近

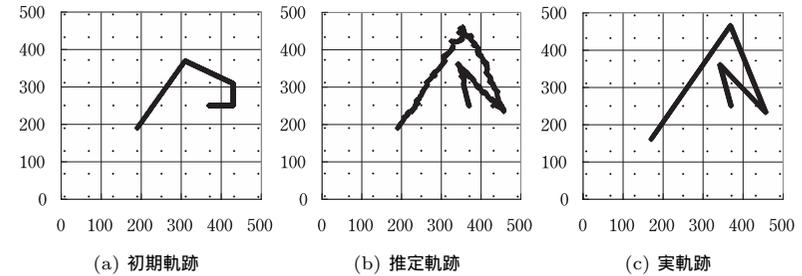


図 7 zigzag シナリオにおける推定過程
 Fig. 7 Estimated trajectories in zigzag scenario.

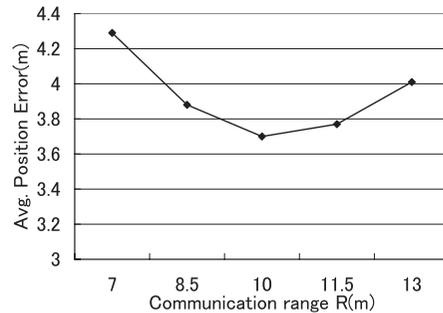
のノードと比べて, 他ノードと遭遇する機会が少ないためである。図 6 (b) から, 中央領域では高精度に移動軌跡を推定できることが分かる。したがって, ノードが他ノードと遭遇する機会が十分に与えられない場合, TRACKIE の移動軌跡の推定誤差が大きくなる傾向にあることが分かる。

また, street walking シナリオと zigzag シナリオの両方の結果より, 逐次修正フェーズを実行した段階での平均位置誤差の値と SA 修正フェーズを実行した段階での平均位置誤差の値とで差がほとんどなく, 逐次修正フェーズのみで十分な精度を達成していることが分かる。図 7 に zigzag シナリオにおける推定例を示す。このノードは, 位置誤差が 4.27 m と平均の値よりも悪い例である。このように平均値よりも悪い例でも, 実軌跡の形状を再現できるため, 多くのノードで軌跡が十分な精度で推定されていると考えられる。

一方, detour シナリオにおいては, 平均位置誤差の値が逐次修正フェーズ後の段階で 7.69 m と大きく, SA 修正フェーズ後の段階で 4.76 m と大幅に減少しており, 同様に平均角度誤差の値も改善されている。これより, zigzag シナリオや street walking シナリオのように, ランドマーク間を直線状の軌跡で移動するノードがある程度存在する場合, 逐次修正フェーズのみで高精度に推定することができ, detour シナリオのようにランドマーク間を直線状の軌跡で移動するノードがまったく存在しない場合でも, 障害物情報などを用いた SA 修正フェーズによりかなりの精度で推定することができることが分かる。

なお, すべてのシナリオにおいて逐次修正フェーズを適用しない場合 (case (iii')) に相当) では高精度に推定することができない。これは, SA 修正フェーズにおいて図 1 (c) のような望ましくない推定となる場合がほとんどとなるためと考えられる。

以上より, street walking シナリオや zigzag シナリオにおけるシミュレーション結果が

図 8 最大通信距離 R による影響Fig. 8 Impact on communication range R .

らランドマーク間を等速直線に近い軌跡で移動するノードがある程度存在する環境では遭遇情報のみを用いて比較的高精度に推定できることが分かる。たとえば、街中のような環境では、ランドマークがいくつかの交差点にあれば、高い確率でランドマーク間を等速直線に近い軌跡で移動するノードが存在するため、このような仮定は妥当である。一方で、等速直線状に移動するノードがまったく存在しない特殊な環境でも、遭遇情報のみである程度の精度を確保することはできる。しかし、street walking シナリオや zigzag シナリオと同様の精度を実現する必要がある場合、障害物情報を必要とする。

5.2 最大通信距離 R の影響

TRACKIE では、遭遇制約を与える際に最大通信距離 R の値を利用している。よって、TRACKIE の精度は最大通信距離 R の値に依存している。しかし、現実環境では、ノードの移動や障害物などによって生じる反射や回折により、無線範囲に揺らぎが生じるため、 R の値を厳密に決定するのは難しい。そこで、無線範囲に揺らぎがある環境において最大通信距離 R が精度に与える影響を評価する。無線の減衰モデルは文献 15) で提案されている Radio Irregularity Model (RIM) を用いる。RIM は無線範囲を 360 度に分割し、揺らぎの程度を表すパラメータ DOI (Degree Of Irregularity) の値に基づき、角度ごとに最小無線到達距離 R_{min} と最大無線到達距離 R_{max} の間を満たすようにランダムに通信距離を決定する。ここでは、 $R_{min} = 7\text{ m}$ 、 $R_{max} = 10\text{ m}$ 、DOI = 0.1 とする。評価シナリオは zigzag シナリオとし、ノードの速度は平均 2.0 m/s、分散 0.1 の正規分布に従うものとする。ノード数は 3,000、ランドマーク数は 81、Hello パケットの送信間隔は 1 秒とする。

評価結果を図 8 に示す。結果より、 $R = 10$ とした場合の誤差が最も小さくなり、 R の値

が 10 m より小さい場合には R の値が小さいほど誤差が大きくなった。これは、ノード間距離が R の値よりも大きい場合でも、Hello パケットが受信されるためと考えられる。また、 R の値が 10 m より大きい場合には R の値が大きいほど誤差が大きくなった。これは、ノード間距離が R よりかなり小さい場合、遭遇制約が十分活かされていないためと考えられる。

5.3 既存手法との比較

我々が調べた限り、オフライン計算により移動軌跡を推定する手法はこれまでに提案されていない。そこで、従来の位置推定手法では利用不可能であるがオフライン計算では利用可能である、対象地理領域内で発生したノード間全遭遇情報の一括利用により、どの程度精度が向上するかを評価するためにリアルタイム方式の位置推定手法と比較する。比較対象として著名なレンジフリーの位置推定手法である MCL⁷⁾ および Amorphous¹⁾ を選択した。MCL は各ノードが自身の存在範囲を保持し、ランドマークの情報を受信するたびにその存在範囲を狭小化する手法である。Amorphous は移動ノードとランドマークにより構成されるアドホックネットワークを介して、ランドマークの情報をマルチホップで伝搬し、各ノードがランドマークとの距離をホップ数により近似する手法である。両手法とも分散型リアルタイムでの位置の推定を対象としているため、集中型オフライン計算で位置を推定する TRACKIE が有利であると思われる。

シミュレーション環境は zigzag シナリオとし、移動速度は平均 2.0 m/s、分散 0.1 の正規分布とする。評価項目は中央 300 m 四方領域の平均位置誤差とする。無線の減衰モデルは RIM とし、ここでは、 $R_{min} = R_{max} - 3\text{ m}$ 、DOI = 0.1 とする。また、各推定手法で想定する最大通信距離 R は R_{min} と R_{max} の平均とする。ノード数、Hello パケット送信間隔、最大無線到達距離、ランドマーク数の 4 つのパラメータについては、うち 1 つのパラメータを表 2 の範囲で変化させ、残りの値を太字の値とする。表 2 に各場合における平均位置誤差の結果を示す。

ノード数 各ノード数における平均隣接ノード数は 3,000, 4,000, 5,000 の場合でそれぞれ 5.16, 6.89, 8.62 となる。表 2 より、どの手法もノード数が多いほど、得られる情報が多くなるため、誤差が改善されることが分かる。特に Amorphous ではノード数が多いほどランドマークからの距離を正確に推定することができるため、誤差が改善されていると考えられる。

Hello パケット送信間隔 表 2 より、いずれの手法も送信間隔が大きいほど、隣接ノードから得られる情報が減少するため誤差が大きくなる。特に、MCL ではランドマークから直接受信した情報のみを用いる手法であるため、得られる情報が他の 2 手法より極端に少な

表 2 TRACKIE, Amorphous, MCL の各パラメータ環境における平均位置誤差
Table 2 Average position errors of TRACKIE, MCL and Amorphous.

Parameter		TRACKIE	MCL	Amorphous
# of Nodes	3,000 4,000 5,000	3.88 3.73 3.74	62.9 61.6 59.4	71.7 55.7 39.8
Hello Packets Interval	1 10 20(s)	3.88 5.25 11.7	62.9 125 162	71.7 71.3 72.5
Wireless Range (R_{max})	10 20 30(m)	3.88 6.57 7.73	62.9 24.8 17.8	71.7 — —
# of Land-marks	57 69 81	4.04 4.20 3.88	111 103 62.9	74.9 78.6 71.7

く、送信間隔による影響が最も大きい。

無線範囲 表 2 より、TRACKIE では無線範囲が大きいほど誤差が増加することが分かる。これは無線範囲が大きいと、遭遇した 2 ノード間の距離制約が緩和されるためである。しかし、いずれの場合も誤差の大きさは $0.4R_{max}$ に抑えられており、十分な精度を達成できているといえる。一方、MCL は無線範囲が大きいほど得られる遭遇エントリが増加するため、精度が飛躍的に向上している。Amorphous についても同様の効果があると考えられるが、無線範囲が 20 m 以上の場合、膨大な量のパケット衝突およびバッファあふれが発生し、ほとんどのノードが位置を推定できなかった。

ランドマーク数 表 2 より、MCL は他の 2 手法と比べて影響が大きいことが分かる。これは MCL はランドマークから直接受信した遭遇情報を必要とする手法であるためである。Amorphous でもランドマークが増加すると精度向上が期待されるが、いずれの場合もランドマークから正確な距離を推定できないため、誤差がそれほど改善されていない。TRACKIE もランドマーク数が大きいほど誤差が改善される傾向にあるが、ランドマーク数が 57 の場合でも十分な精度を達成しているといえる。

以上より、提案方式では、平均位置誤差が約 $0.4R_{max}$ 程度となり、既存のリアルタイム推定手法よりも誤差が小さい。また、文献 16) で述べられているように、GPS の誤差は数 m から数十 m となるのに対し、提案方式はシミュレーション結果より約 5 m の誤差を実現できる。よって、提案方式は十分有用であるといえる。

6. まとめ

本論文では低コストかつ容易に実現可能で、高精度にノードの移動軌跡を推定するアルゴリズム TRACKIE を提案した。提案手法ではノード間の遭遇情報を用いて、逐次推定およびシミュレーテッド・アニーリングという 2 段階のフェーズにより移動軌跡を修正する。シミュレーション結果より現実的な環境下で推定誤差が最大無線到達距離の 40% 程度に抑えられることを確認した。今後、提案手法を MOTE に実装し、その有用性についても示す予定である。

参考文献

- 1) Nagpal, R., Shrobe, H. and Bachrach, J.: Organizing a global coordinate system from local information on an ad hoc sensor network, *Proc. IPSN 2003*, pp.333–348 (2003).
- 2) MobiREAL: MobiREAL web page (online). available from <http://www.mobireal.net>
- 3) Maeda, K., Sato, K., Konishi, K., Yamasaki, A., Uchiyama, A., Yamaguchi, H., Yasumoto, K. and Higashino, T.: Getting urban pedestrian flow from simple observation: realistic mobility generation in wireless network simulation, *Proc. MSWiM 2005*, pp.151–158 (2005).
- 4) Bahl, P. and Padmanabhan, V.N.: RADAR: an in-building RF-based user location and tracking system, *Proc. INFOCOM 2000*, pp.775–784 (2000).
- 5) Priyantha, N.B., Chakraborty, A. and Balakrishnan, H.: The Cricket location-support system, *Proc. MobiCom 2000*, pp.32–43 (2000).
- 6) Bulusu, N., Heidemann, J. and Estrin, D.: GPS-less low cost outdoor localization for very small devices, *IEEE Personal Communications*, Vol.7, No.5, pp.28–34 (2000).
- 7) Hu, L. and Evans, D.: Localization for mobile sensor networks, *Proc. MobiCom 2004*, pp.45–57 (2004).
- 8) Uchiyama, A., Fujii, S., Maeda, K., Umedu, T., Yamaguchi, H. and Higashino, T.: Ad-hoc localization in urban district, *Proc. INFOCOM 2007 Mini-Symposium*, pp.2306–2310 (2007).
- 9) Shang, Y., Ruml, W., Zhang, Y. and Fromherz, M.P.J.: Localization from mere connectivity, *Proc. MobiHoc 2003*, pp.201–212 (2003).
- 10) Brooks, R.R., Ramanathan, P. and Sayeed, A.M.: Distributed target classification and tracking in sensor networks, *Proc. IEEE*, Vol.91, No.8, pp.1163–1171 (2003).
- 11) Zhao, F., Liu, J., Liu, J., Guibas, L. and Reich, J.: Collaborative signal and in-

- formation processing: An information directed approach, *Proc. IEEE*, Vol.91, No.8, pp.1199–1209 (2003).
- 12) Taylor, C., Rahimi, A., Bachrach, J., Shrobe, H. and Grue, A.: Simultaneous localization, calibration, and tracking in an ad hoc sensor network, *Proc. IPSN 2006*, pp.27–33 (2006).
- 13) Corana, A., Marchesi, M., Martini, C. and Ridella, S.: Minimizing multimodal functions of continuous variables with the “Simulated Annealing” algorithm, *ACM Trans. Mathematical Software*, Vol.13, No.3, pp.262–280 (1987).
- 14) Google Maps: Google Maps web page (online). available from <http://maps.google.co.jp>
- 15) Zhou, G., He, T., Krishnamurthy, S. and Stankovic, J.A.: Models and solutions for radio irregularity in wireless sensor networks, *ACM Trans. Sensor Networks*, Vol.2, No.2, pp.221–262 (2006).
- 16) 岡峰 正, 國頭吾郎, 高橋竜男, 田中 聡: GPS 携帯電話による位置情報ログを用いた訪問施設抽出方法, マルチメディア, 分散, 協調とモバイル (DICOMO 2006) シンポジウム, pp.273–276 (2006).

(平成 19 年 12 月 31 日受付)

(平成 20 年 7 月 1 日採録)

推薦文

本論文は, GPS などの測位システムを利用せず, 少数の位置基準点と無線端末が蓄積した他端末との総合情報のみに基づき端末の移動軌跡を推定する低コストな位置トレース手法を提案している.

提案手法では, 与えられた遭遇情報から位置基準点間の移動がなるべく最短距離に近い端末の行動を検出し, それを基準とした逐次的推定を行うことで, 高速かつ簡潔な位置トレーシングを実現している.

また, シミュレーテッド・アニーリングを利用することでさらなる精度の向上をシミュレータ上にて実証している.

本研究成果をもとに, 衛星電波が届かないビル街および地下街などにおける移動体端末の位置補正技術の高度化と人車間における衝突事故防止技術への応用を可能とし, 安全・安心な交通環境を提供することが期待できる.

よって, 本論文を推薦論文として推薦します.

(高度交通システム研究会主査 小花貞夫)



藤井 彩恵 (学生会員)

平成 20 年大阪大学大学院情報科学研究科情報ネットワーク学専攻博士前期課程修了. 同年同大学院博士後期課程進学. 平成 20 年より日本学術振興会特別研究員. アドホックネットワークに関する研究に従事.



内山 彰 (学生会員)

平成 20 年大阪大学大学院情報科学研究科情報ネットワーク学専攻博士後期課程修了. 現在, 日本学術振興会特別研究員. 博士 (情報科学). ネットワークセキュリティやアドホックネットワークの研究に従事.



梅津 高朗 (正会員)

平成 13 年大阪大学大学院基礎工学研究科情報数理系専攻博士前期課程修了. 同年同大学院博士後期課程進学. 平成 14 年同大学院博士後期課程退学後, 同大学院情報科学研究科助手. 平成 19 年より同大学院情報科学研究科助手. 博士 (情報科学). アドホックネットワーク用モデルウェアや開発環境の研究に従事.



山口 弘純 (正会員)

平成 6 年大阪大学基礎工学部情報工学科卒業. 平成 10 年同大学大学院基礎工学研究科博士後期課程修了. 同年オタワ大学客員研究員. 平成 11 年大阪大学大学院基礎工学研究科助手. 平成 14 年同大学院情報科学研究科助手. 平成 19 年より同大学院情報科学研究科准教授. 博士 (工学). 分散システムや通信プロトコルの設計および実装に関する研究に従事. IEEE, 電子情報通信学会各会員.



東野 輝夫 (フェロー)

昭和 54 年大阪大学基礎工学部情報工学科卒業。昭和 59 年同大学大学院基礎工学研究科博士後期課程修了。同年同大学助手。現在、同大学大学院情報科学研究科教授。平成 19 年 10 月より、独立行政法人科学技術振興機構、CREST、研究代表者。博士 (工学)。分散システム、通信プロトコル、モバイルコンピューティング等の研究に従事。電子情報通信学会、

ACM 各会員。IEEE Senior Member。
