

少ない基準位置情報を移動無線端末間で補完する 都市部向け位置推定手法の提案

藤井 彩恵 内山 彰 前田 久美子 梅津 高朗 山口 弘純 東野 輝夫

大阪大学 大学院情報科学研究科

本稿では都市部における移動端末のための位置推定手法を提案する。ビルが林立する地区や地下街が存在する都市部では GPS 情報取得が困難である。このため既存手法の多くは、常に正確な位置を把握できるノード（シード）が発信する基準位置情報をアドホックネットワークで伝播させることにより、シードからの距離を推定する手法をとる。しかし都市部では歩行者や車両の移動によりアドホックネットワークが流動的であり、頻繁に分断されるといった問題がある。提案手法では、シードが発信する基準位置情報に加え、各移動ノードは、遭遇した他の移動ノードが周囲に発信する推定位置情報も自身の位置推定に利用する。また、道路の形状など対象領域の移動可能領域を各移動ノードに与え、移動可能領域における実移動距離を考慮した位置推定を行うことにより、精度を向上させる工夫もしている。シードの密度や位置情報の送信間隔による精度の変化などをシミュレーションにより評価し、シードが建物や路側に数十～百メートルの疎な間隔で設置されているような状況でも、位置推定誤差が無線到達距離と同程度以下におさえられることを確認した。

A Proposal of Localization Algorithm with Sparsely Deployed Seeds in Urban District

Sae Fujii Akira Uchiyama Kumiko Maeda Takaaki Umedu
Hirozumi Yamaguchi and Teruo Higashino

Graduate School of Info. Sci. & Tech., Osaka Univ., Japan

In this paper, we propose a localization algorithm for mobile nodes in urban district. In most localization techniques, each node uses the location information sent from seed nodes. Then, ad hoc networks assist to deliver the location information and to measure the distance from the seeds. However, mobile nodes cannot always receive the location information because ad hoc networks are frequently partitioned due to node movement and obstacles such as buildings. In our method, each node uses location information sent from its neighboring mobile nodes. Moreover, we employ obstacle information to calculate the movable areas of each mobile node more accurately. We have evaluated the performance of our method by simulation with sparsely deployed landmarks, and have confirmed that the average position estimation error was less than the radio range.

1 はじめに

交通ナビゲーションシステムや位置依存サービスなどをはじめとした位置情報の活用が期待されている。このため、ノードの存在位置を計測・推測するための位置推定手法がこれまでに数多く提案されている。

一般に、位置推定手法ではシードとよばれるいくつかのノードが提供する正確な位置情報やシードからの距離情報を用いて各ノードの位置を決定する。シードからの情報を伝える方法として、長距離無線機器と少数のシードを仮定する手法や、RFID や電波タグのような安価な短距離無線機器によるシードを多数配置する手法が考えられる。前者の手法では離れたシードから無線信号を受信するため見通しが良くなければならず、建物や地下街といった見通しの悪い場所が多い都市部では難しい。一方、後者の手法では通信距離の短いシードを多数配置する必要があり、コスト面で問題がある。文献 [1] では移動するシードを仮定することにより、シードの位置情報を伝播可能な領域を拡張している。しかし、移動ノードによる位置推定の精度は、領域の広さや移動ノード密度に大きく依存する。このため、特に低速移動する歩行者を想定する場合、ノードが移動シードを頻繁に発見することは困難となり、シードからの位置情報が十分に得られないとい

う問題がある。

また、いくつかの手法では、シードからの間接的な情報を利用する。これらの手法は、文献 [2] を参考に次の二つの手法に分類できる。協調的多点測定 (collaborative multi lateration) に分類される手法 (例えば文献 [3]) では、無線アドホックネットワーク上でシードの位置情報をマルチホップ伝播し、シードへの距離をホップ数などで近似した上で位置推定を行う。反復的多点測定 (iterative multi lateration) に分類される手法 (例えば文献 [4]) では、あるノードの推定位置からその隣接ノードの位置を推定する処理を繰り返す。しかし、都市部での歩行者や自動車は静止しているわけではなく、ビルのような障害物やノードの不均等な配置・密度の影響により、常に無線ネットワークが変化し頻繁に分断される可能性がある。したがって、これらの手法は都市部での利用には適さない場合が多い。

本稿では都市部における移動ノードのための位置推定手法 UPL (Urban Pedestrians Localization) を提案する。UPL では、シード情報を直接受信する機会が少なく、かつ障害物などにより無線アドホックネットワークが分断されやすい都市部での利用を想定する。各移動ノードは自身の推定存在領域を常に保持する。時間が経過すると移動可能領域が拡大す

るため、推定存在領域も拡大する。そこで UPL では推定存在領域を縮小させるために、遭遇する機会の少ないシードからの位置情報だけでなく、他の移動ノードの位置情報も利用する。他の移動ノードと遭遇した場合には、互いの推定存在領域情報を交換し、遭遇ノードから得た推定存在領域及び遭遇ノードからの無線の到達領域の情報に基づいて自身の推定存在領域を更新する。情報を受信できた事実から、遭遇したノードの推定存在領域に電波到達距離を加算した領域に自身が存在することが分かるため、その領域と自身の推定存在領域との共通領域を自身の推定存在領域とすることができる。したがって、UPL ではネットワークが分断され、シードの位置情報が伝播されない場合も位置推定が可能である。また、提案手法では都市部における建造物などの障害物情報を利用してノードの推定存在領域をさらに正確に計算する工夫もしている。なお、この障害物情報はシードから位置情報とともに配布されるものとする。

提案手法の性能評価のため、ネットワークシミュレータ MobiREAL[5, 6, 7] を用いてシミュレーションを行った。実験結果より、UPL は JR 大阪駅前における道路地形、歩行者の密度・行動経路を再現した環境において、無線到達距離 10m、平均シード配置間隔を約 70m とした場合、その位置推定誤差は平均で約 8m 程度であることを確認した。これにより UPL は一定の端末普及率のもとで、歩行者ナビゲーション、位置依存情報取得や安全安心のための児童追跡システムなど、都市部での利用が想定される位置情報サービスに十分適用可能な精度の位置情報を提供できることが分かった。

2 UPL の概要

歩行者は ZigBee や Bluetooth といった何らかの PAN 通信デバイスを装備した端末を保持しているものと想定する。これは、近年の Bluetooth 搭載の携帯電話の増加や ZigBee ハードウェアの小型化を考慮した場合、現実的な仮定といえる。以降、議論の簡単のため、全てのノードの無線到達距離は同じものとし、これを r で表す。また、ノードの最大速度を V_{max} で表す。時刻 t においてノード i が存在すると推測される領域（推定存在領域、以下単に存在領域とよぶ）を R_i^t とする。

シードを含む各ノードはその隣接ノードに対し、一定の間隔で Hello メッセージとよばれる情報通知メッセージをブロードキャストする。ノード i がメッセージを送信した時刻を t で表すと、ノード i が時刻 t に送信した Hello メッセージにはノード i が最後に位置推定を行った時刻 t_i から時刻 t までの経過時間 $\Delta t_i = t - t_i$ と存在領域 $R_i^{t_i}$ の情報が含まれるものとする。ノード i がシードであれば、 Δt_i は 0、 $R_i^{t_i}$ はシードの正確な位置座標である。 Δt_i はノード i が保持するタイムによって計算されるため、ノード間の時刻同期は不要である。また、Hello メッセージには対象領域の障害物地図が含まれる。障害物地図はシードから散布され、Hello メッセージにより他ノードに伝搬されるものとする。存在領域と障害物地図のデータ構造は 3.1 節で説明する。

シード以外のノード i がノード j からの Hello メッセージを受信した際、ノード i は即座に UPL アルゴリズムを実行し、自身の存在領域を狭小化する。ノード i が Hello メッセージを受信した時刻を t とすると、この Hello メッセージにはノード j が時刻 t_j に更新した（狭小化した）存在領域 $R_j^{t_j}$ と時刻 t_j からの経過時間 Δt_j が含まれる。UPL アルゴリズムでは、まずノード i は自身の存在領域 $R_i^{t_i}$ と Hello メッセージに含まれる $R_j^{t_j}$ 及び障害物地図 M (図 1 (a)) から、それぞれ R_i^t と R_j^t を計算し、さらに R_i^t を基に、時刻 t においてノード j からの電波を受信可能な領域を計算する (図 1 (b) 及び (c))。この領域は、 R_j^t を無線到達距離 r だけ拡張することにより計算され、以下 $R_j^t \oplus r$ で表す。ノード i は R_i^t と $R_j^t \oplus r$ の両方に含まれるため、 R_i^t と $R_j^t \oplus r$ の

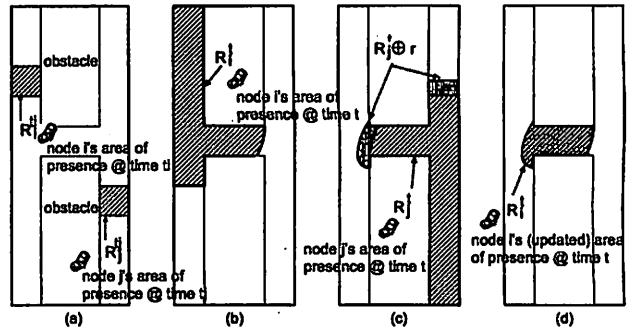


図 1: UPL アルゴリズム

共通領域を、時刻 t におけるノード i の存在領域とする。この操作により存在領域が狭小化される (図 1 (d))。

ここで、 R_i^t が時刻 t におけるノード i の位置を含むとき、かつそのときに限り R_i^t は完全であるという。さらに、時刻 t におけるノード i の 2 つの完全な存在領域 R_i^t と \hat{R}_i^t に対し、 $|R_i^t| < |\hat{R}_i^t|$ ならば、 R_i^t は \hat{R}_i^t よりも正確であるという (ただし、 $|R_i^t|$ は R_i^t の面積を表す)。本研究の目的は、可能な限り完全で、かつ正確なノードの存在領域を決定するアルゴリズムを設計することである。

存在領域の情報は多くのサービスに利用することができる。例えば、サービスプロバイダが顧客の存在領域を受信し、その地区のお店やレストランについての情報を顧客に送信する場合、なるべく完全かつ正確な存在領域を決定できることは極めて重要である。一方、ナビゲーションシステムなど、ノードの推定位置を提示する必要があるサービスでは、3.4 節で述べるような、存在領域の中で何らかの基準により最も存在する可能性が高い位置を決定する位置決定関数を用いる。

3 UPL アルゴリズムの詳細設計

3.1 存在領域と障害物地図のデータ構造

既存の位置推定手法では、存在領域を表すために様々なデータ構造が使用されている。文献 [4] では、そのうちいくつかのデータ構造について詳細に論じている。最も単純な方法は、円や長方形などの多角形を用いる方法である。しかし円や多角形では存在領域を正確に表すことはできない。MCL[1] では、存在領域を表現するために存在領域内からランダムに選択された点の集合を利用する。Sextant[4] は複数の代表点間のベジエ曲線により存在領域を表す。

提案手法では障害物間の複雑な形状の領域におけるノードの移動を考慮するため、存在領域の形状が既存手法よりも複雑となる。このため、対象領域を小さなグリッドに分割し、グリッドの集合によって存在領域や障害物を表す簡潔なデータ構造を採用し、位置推定アルゴリズムの軽量化を図る。なお、単位グリッドの大きさによってはデータ量の増大が懸念されるが、これについては 5 章で論じる。

3.2 アルゴリズムの説明

存在領域の計算 ノード j が最後に位置推定を行ってからの経過時間 $\Delta t_j = t - t_j$ 、グリッドの集合によって表される $R_j^{t_j}$ ($t_j < t$) 及び障害物地図 M が与えられた際に、時刻 t におけるノード j の存在領域 R_j^t を計算するアルゴリズムを示す。以降、このアルゴリズムを APC (Area of Presence Computation) アルゴリズムとよぶ。

APC アルゴリズムの概要は以下の通りである。障害物地図 M の移動可能な領域内の全グリッド集合を FS で表す。 R_j^t を計算するために、 $R_j^{t_j}$ からの距離が $V_{max} \cdot \Delta t_j$ 以内である FS のグリッドを R_j^t に加える。しかし、障害物が存在する場合、 FS 内の各グリッドごとに $R_j^{t_j}$ からの正確な最短

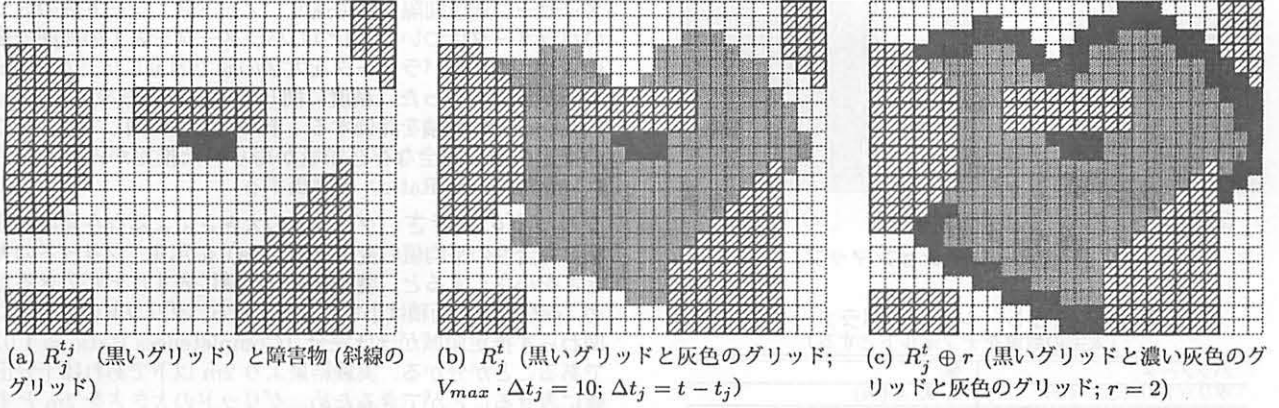


図 2: 存在領域の計算と受信可能な領域の計算 (グリッドの大きさ $w = 1$)

距離を計算することは複雑である。したがって、ここでは最短距離として近似値を用いる代わりに、 R_j^t にグリッドを逐次追加していくだけで R_j^t が求められる軽量なアルゴリズムとして APC を実現する。

ここで、単語と表記の定義を与える。グリッド g に対して g と一辺を共有するグリッドを辺接グリッド (side grid) とし、 g の辺接グリッド以外で、一頂点のみを共有するようなグリッドを点接グリッド (diagonal grid) とよぶ。また、辺接グリッドと点接グリッドを合わせて隣接グリッド (neighboring grid) とよび、グリッドの集合 G 内の各グリッドに対し、その隣接グリッドのうち少なくとも一つが G に含まれないようなグリッドを境界グリッド (border grid) とする。グリッド g に対し、 $d(g)$ を R_j^t からの最短距離とする。すなわち、 g が R_j^t 内に含まれるならば $d(g) = 0$ となる。隣接する二つのグリッド間の距離はグリッドの中心点同士のユークリッド距離とする。

APC では、はじめに R_j^t の初期値を R_j^t とする。次に w をグリッドの一辺の長さとする、 R_j^t の各境界グリッド g に対し、 $d(g) + w \leq V_{max} \cdot \Delta t_j$ ならば、 FS に含まれる g の辺接グリッド g' を R_j^t に加える。 g' を R_j^t に加えた場合、 $d(g') = d(g) + w$ とする。同様に、 g の点接グリッド g'' に対して、 $d(g'') + \sqrt{2}w \leq V_{max} \cdot \Delta t_j$ ならば R_j^t に加える。 g'' を加えた場合、 $d(g'') = d(g) + \sqrt{2}w$ とする。そして、 R_j^t に加えられた g' と g'' を FS から削除する。以上の手続きを R_j^t へ加えることのできる FS のグリッドが無くなるまで繰り返す。

図 2 に R_j^t と R_j^t 、 $R_j^t \oplus r$ の例を示す。また、APC アルゴリズムの記述は図 3 のとおりである。

受信可能な領域の計算 時刻 t におけるノード j の存在領域 R_j^t を無線到達距離 r だけ拡張することによりノード j の通信を受信可能な領域 $R_j^t \oplus r$ を計算する。これには、前述の APC アルゴリズムが利用できる。なお、都市部での電波の伝播は、自由空間と比較して減衰や回折の影響を大きく受けるため、これらの現象を考慮したアルゴリズムを設計することが精度の観点からは理想である。しかし、携帯端末など処理能力に制約がある環境では複雑なアルゴリズムを実装することは困難であり、かつバッテリー容量の観点からも得策ではない。また、通信距離が短い PAN デバイスを仮定した場合、それらの物理現象の影響は少ない。したがって、受信可能な領域を計算するアルゴリズムとして、 R_j^t から距離 r 以下でかつ R_j^t のいずれかのグリッドから直線で見通せるグリッドのみを $R_j^t \oplus r$ に追加するように変更した APC アルゴリズムを用いる。

共通領域の計算 R_j^t と $R_j^t \oplus r$ の共通領域は両方の領域に含まれるグリッドを求めることで得られる。以上より新しい存在領域 $R_j^t \cap \{R_j^t \oplus r\}$ が求められる。

```

1  $R_j^t = R_j^t; B_0 = \text{set of border grids of } R_j^t;$ 
2  $FS = \text{set of grids in free space except } R_j^t; h = 0;$ 
3 do {
4    $B_{h+1} = \emptyset;$ 
5   for each  $g \in B_h$  {
6     for each side grid  $g'$  of  $g$  in  $FS$  {
7       if  $(d(g) + w \leq V_{max} \cdot \Delta t_j)$  then {
8          $B_{h+1} = B_{h+1} \cup \{g'\}; d(g') = d(g) + w;$ 
9          $FS = FS - g';$ 
10      }
11    }
12   for each diagonal grid  $g''$  of  $g$  in  $FS$  {
13     if  $(d(g) + \sqrt{2}w \leq V_{max} \cdot \Delta t_j)$  then {
14        $B_{h+1} = B_{h+1} \cup \{g''\}; d(g'') = d(g) + \sqrt{2}w;$ 
15        $FS = FS - g'';$ 
16     }
17   }
18 }
19  $R_j^t = R_j^t \cup B_{h+1}; h = h + 1;$ 
20 } while  $(B_h \neq \emptyset)$ 

```

図 3: APC アルゴリズム

3.3 精度向上のための工夫

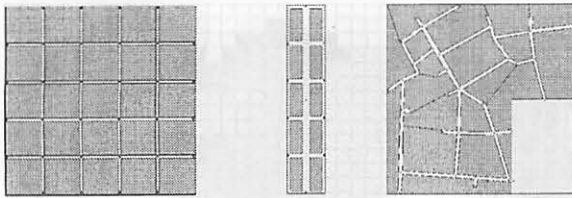
いくつかの既存手法では、より精度の高い位置推定を行うための様々な工夫がなされている。UPL では、MCL[1] で提案されている二つの手法を採用した。一つは、シードからのメッセージ受信履歴を用いる手法である。受信履歴から、ノードがシードの無線到達領域への出入りを検出できれば、そのノードがシードの無線到達領域の境界付近に存在することが分かる。もう一つは、シードの位置情報を 2 ホップ伝播させる手法である。この手法では、シードからの位置情報を直接受信したノードが隣接ノードに再度その位置情報を送信する。2 ホップで情報を受信したノードは 1 ホップで受信できる領域に存在しておらず、かつ 2 ホップで受信できる領域に存在していることが分かるため、これを利用したさらなる狭小化を行うことができる。

3.4 位置決定関数

存在領域からノードの位置を決定する場合を考慮し、存在領域の中で存在する可能性が最も高い点を決定する位置決定関数を提案する。存在領域 R と障害物地図 M から、存在する可能性が最も高い点 p を決定するために、次の関数を用いる。

$$\text{select } p \in R \text{ that minimizes } \max_{p' \in R} \text{dist}(p, p') \quad (1)$$

$\text{dist}(p, p')$ を障害物地図 M 上での点 p と点 p' 間の最小距離 (障害物による移動不可能領域を考慮した上での最小距離) とする。 R が完全であれば、そのノードの実際の存在位置が R 内に存在する。ここで、後述する実験結果より、提案手法に



(a) マンハッタン (b) 中央分離帯 (c) 繁華街

図 4: シミュレーションマップ

表 1: シミュレーションのパラメータ
(太字の数値をデフォルトとする)

パラメータ	値
グリッドの大きさ (w)	1, 2 , 4 (m)
Hello メッセージの送信間隔	1, 2 , 5, 10 (秒)
移動速度の分布	[1.0, 2.0], [3.0, 4.0], [5.0, 6.0], [7.0, 8.0], [9.0, 10.0] の正規分布
ノード数	500, 1,000, 2,000 , 3,000
シード数	8, 12, 16
無線到達距離 (r)	10 (m)

おける存在領域は高い確率（おおよそ 98 %）で完全であることから、 R の内部の点 p と R 内の他の点 p' 間の最大距離を最小とするような点 p を選択すれば、位置推定誤差は高い確率で最小になると考えられる。なお、 R 内のすべての点を対象にすると計算コストが増大するため、 R に含まれる各グリッドの中央点のみを点 p の候補とする。

4 性能評価

4.1 シミュレーション設定

シミュレーションでは、格子状道路のマップ（マンハッタンのマップ）[8]（図 4 (a)）、中央分離帯のマップ（図 4 (b)）、繁華街のマップ（図 4 (c)）の計 3 つのマップを使用する。マンハッタンのマップは $500m \times 500m$ の領域で、幅 8m の道路を 8 本持ち、その他の領域は移動も通信も遮断する障害物である。中央分離帯のマップは $100m \times 500m$ の領域で、道路と障害物が配置されている。ただし、マップ中央の道路の中央には中央分離帯が設置されている。中央分離帯とは、障害物とは異なり、移動は遮断するが通信は可能であるような領域である。繁華街のマップは大阪駅前をモデルとした $500m \times 500m$ のマップであり、そのうち南東 $170m \times 250m$ は自由空間とした。いずれのマップを使用する場合も、ノードははじめに一緒に配置され、各ノードは、道路上では交差点で後方を除く方向のうちランダムに行き先を決定する Random Street Decision モビリティモデルに沿って移動し、自由空間では Random Waypoint モビリティモデル [9] に沿って移動する。

本稿では、3 種類のシミュレーション実験の結果を示す。一つ目の実験では、UPL の性能特性の評価として、様々なパラメータを変化させた際の精度（存在領域の正確性）と完全性への影響を評価する。図 3 の APC アルゴリズムに従えば、理論上、推定結果は完全でなければならない。しかし APC アルゴリズムを実装する際には、存在領域をグリッドによる表現に近似する必要がある。このとき理論上の存在領域とグリッドによって表現された実際の存在領域との間に誤差が生じ、存在領域は必ずしも完全ではなくなる。よって精度のみならず完全性も評価の対象とした。二つ目の実験では各ノードの存在領域情報及び障害物を考慮した移動可能領域の計算が精度と完全性に与える影響を評価する。三つ目の実験では、既存手法 Amorphous[3] と MCL[1] との比較を行う。

4.2 パラメータの影響

ここでは、パラメータを変化させた場合の完全性と精度に与える影響を評価する。シミュレーションにはマンハッタンのマップを用いた。表 1 に示すグリッドの大きさ、Hello

メッセージ送信間隔、移動速度、ノード数、シード数の 5 つのパラメータについて、1 つのパラメータを表 1 の範囲で変化させ、残りのパラメータを太字の値で固定した上でシミュレーションを行った。精度に関しては、存在領域（Area of Presence）の面積を評価する。また完全性に関しては、全ての推定のうち完全な存在領域が得られた推定が占める割合（Completeness Ratio）を評価する。

グリッドの大きさ グリッドの大きさによる存在領域の面積と完全性の平均値の変化を図 5 (a) に示す。グリッドの大きさを小さくすると、境界をより正確に表すことができるため、存在領域の面積は小さくなる。一方、グリッドの大きさに関わらず推定領域がほぼ完全（Completeness Ratio = 1.0）であることが分かる。実験結果より $2m$ 以下であれば十分正確に表せることができるため、グリッドの大きさを $2m$ とするのが適当であると判断できる。

Hello メッセージ送信間隔 送信間隔による存在領域の面積と完全性の平均値の変化を図 5 (b) に示す。送信間隔が長くなると得られる情報が少なくなるため、存在領域の面積は大きくなり、精度は悪化する。例えば、移動ノード i がシード j の近傍を直線的に通過する場合（シード j からのメッセージを受信するためには、その送信間隔は $\frac{2r}{V_{max}}$ でなければならない。 $r = 10m$, $V_{max} = 2$, 送信間隔が 10 秒の場合、ノード i がシード j の近傍を通過したとしても Hello メッセージを受信しない可能性があるため、実験結果からも存在領域は大きくなる。一方、送信間隔に関わらず推定領域はほぼ完全（Completeness Ratio = 1.0）であった。実験結果より送信間隔が短いほど精度が良いという結果が得られたが、送信間隔が短すぎるとメッセージ数が多くなる点や送信間隔が 1 秒の場合と 2 秒の場合で存在領域は大きく変わらない点を考慮すると、送信間隔を 2 秒とするのが適当であると判断できる。

移動速度 移動速度による存在領域の面積と完全性の平均値の変化を図 5 (c) に示す。特にノードの移動速度が変化した場合の影響を評価するために、実験では最小速度と最大速度を変化させた。この結果から存在領域の面積は速度に対して線形に増加することが分かる。各ノードの存在領域は移動速度に応じて広がるため、線形に増加すると考えられる。一方、移動速度が大きいほど存在領域をグリッドで近似する際の誤差が大きくなるため、完全性は悪化する。これより、提案手法は低速度で移動する歩行者の位置推定に適しているといえる。

ノード数 ノード密度による存在領域の面積と完全性の平均値の変化を図 5 (d) に示す。ノード密度が大きいほど、より多くのノードと遭遇するため、得られる情報が増え、存在領域は小さくなる。一方、ノード密度に関わらず推定領域がほぼ完全（Completeness Ratio = 1.0）であることが分かる。これより、精度を向上させるためにはある程度のノード密度が必要となる。ただし、これはある程度の移動ノードに遭遇する必要があることを意味しており、移動ノード間で常にアドホックネットワークを形成しておく必要があるわけではない。

シード数 シード密度による存在領域の面積と完全性の平均値の変化を図 5 (e) に示す。この結果から、シード数が増えるにつれ、精度が向上しているのが分かる。しかし 1 個のシードが直接情報を送信できる領域は無線到達距離が $10m$ であることから $10m \times 8m \times 4 = 320m^2$ であるため、16 個のシードが直接情報を送信できる領域は全移動可能領域（約 $32000m^2 = 8 \times 8m \times 500m$ ）のうち 16 % である。このようにシードがまばらに配置された状況でも、存在領域の面積は約 $200m^2$ であり¹、UPL は高い精度を達成しているとい

¹ 例えば幅 8m の道路であれば、道路上の長さ 25m の範囲内に存在すると推定されている

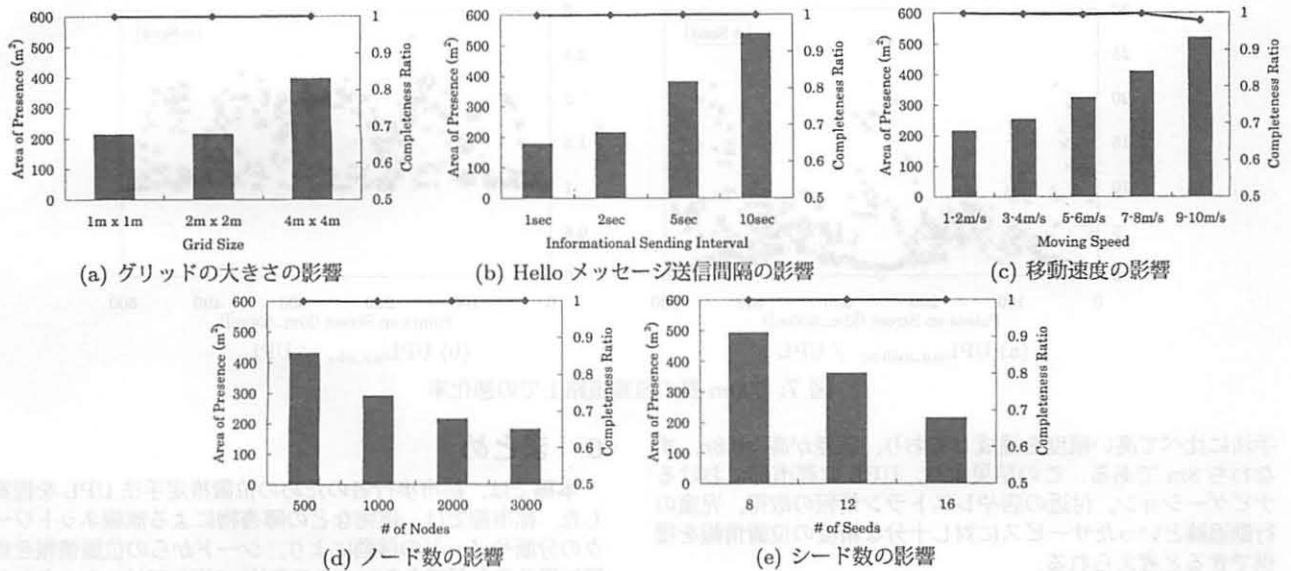


図 5: パラメータ値が UPL の性能に与える影響 (棒グラフ: Area of Presence, 折れ線グラフ: Completeness Ratio)

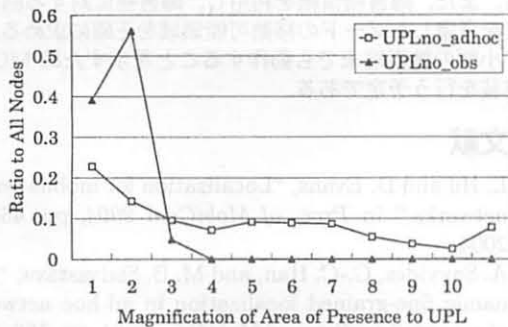


図 6: 悪化率の分布

える。

4.3 位置推定の精度と完全性

ここでは、各ノードの存在領域情報及び障害物を考慮したノード移動範囲の計算により、位置推定の完全性や精度がどの程度向上しているかを評価する。そのために、遭遇する移動ノードの位置情報を利用しない UPL_{no_adhoc} とノードの移動範囲計算時に障害物情報を利用しない UPL_{no_obs} という UPL の 2 つの簡易版で存在領域の面積や完全性を測定した。実験は中央分離帯のマップで行い、パラメータはノード数 400、シード数 11、Hello メッセージ送信間隔を 5 秒とし、それ以外は表 1 の太字の値とした。

まず、同条件の UPL に対して推定精度がどの程度低下したかを悪化率 (存在領域の増大率) とよび、この悪化率 (> 1.0) をノードごとに測定した。悪化率を X 軸にとり、その悪化率であったノード数が全ノード数に占める割合を Y 軸にとった分布図を図 6 に示す。図 6 から移動ノードの位置情報は多数のノードの大幅な精度向上に寄与していることが分かる。一方、障害物情報を使用しない場合、半数以上のノードの面積が約 2 倍となることから、移動ノードの位置情報ほどではないが、障害物情報も精度向上に寄与していることがわかる。

同様に 500m 長の直線道路に着目し、その道路上におけるノード位置を X 軸にとり、対応する悪化率を Y 軸にプロットしたグラフを図 7 に示す。図 7 (a), (b) は、それぞれ道路上の各座標ごとに UPL に対する UPL_{no_adhoc} , UPL_{no_obs} の存在領域の面積の悪化率を示す。シードは、0m, 250m, 500m の位置に存在している。図 7 (a) から、遭遇するノードの位置情報を利用しない場合、ほとんどのノード、特にシードから離れた場所にいるノードで、悪化率が大幅に増大して

表 2: 悪化率と完全性の割合

	悪化率	完全性の割合
UPL_{no_adhoc}	4.529	1.0
UPL_{no_obs}	1.274	1.0
UPL	1.000	1.0

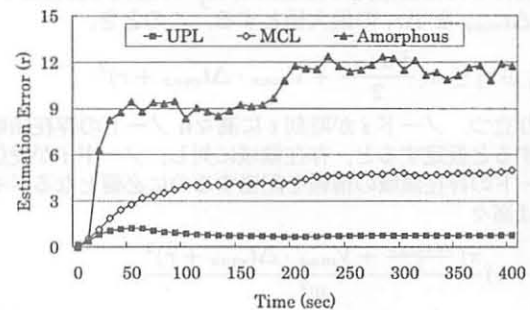


図 8: 位置推定の平均誤差

いることが分かる。また、図 7 (b) から、障害物情報を使用しない場合、各交差点の中間点付近のノードで悪化率が増大する傾向があることが分かる。

また、表 2 は悪化率と完全性の平均値の結果を示している。表 2 から、いずれの場合も完全であり、他ノードの位置情報や障害物情報により、それぞれ 77%, 21% の存在領域の面積の平均値を削減していることが分かる。以上の結果から、UPL は他ノードの位置情報及び障害物情報により大幅な精度向上を実現していることが確認できた。

4.4 位置推定誤差

他手法との比較を行うために、繁華街のマップを用いて UPL, MCL[1], Amorphous[3] の位置推定誤差を測定した。シードは各交差点に配置した。このとき、平均シード配置間隔は約 70m となる。それ以外のパラメータは表 1 のデフォルト値とした。図 8 では無線到達距離 ($r = 10m$) に対する位置推定誤差 (10 回のシミュレーションの平均値) の時間変化を示す。図 8 から Amorphous は比較的精度が低いことが分かる。これに対し MCL は、シードから 2 ホップ以内で受信した情報のみを使用しているにも関わらず、Amorphous より性能が良い。これは多くの領域が障害物によって制限されるような都市部において、ホップ数に基づき高精度な距離推定を行うことが難しいためである。一方、UPL はこれらの

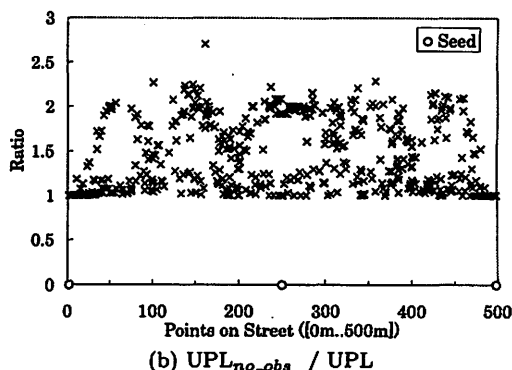
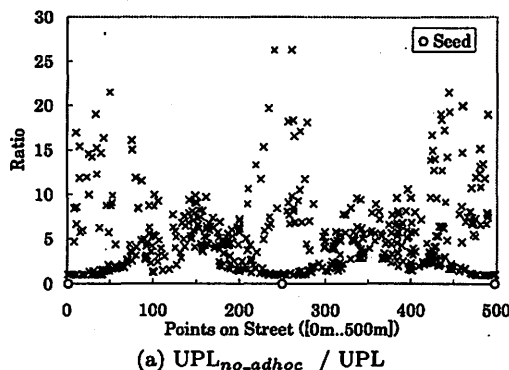


図 7: 500m 長の直線道路上での悪化率

手法に比べて高い精度を達成しており、誤差が高々 $0.8r$ 、すなわち $8m$ である。この結果より、UPL は都市部におけるナビゲーション、付近の店やレストラン情報の取得、児童の行動追跡といったサービスに対し十分な精度の位置情報を提供できると考えられる。

5 メモリ使用量と時間計算量

APC アルゴリズムを実装する際に要するメモリ使用量と時間計算量について議論する。

はじめに、存在領域を無線到達距離分拡張した領域の大きさ $|R_j^t \oplus r|$ を解析する。障害物が全く存在しない最悪の場合について考える。 dia_{max} を時刻 t_j においてノード j が最後に実行した位置推定の推定領域 $R_j^{t_j}$ の最大の直径とする。また Δt_{max} を Δt_j の最大値とする。このとき、

$$|R_j^t \oplus r| \leq \pi \left(\frac{dia_{max}}{2} + V_{max} \cdot \Delta t_{max} + r \right)^2$$

が成り立つ。ノード i が時刻 t に高々 n ノードの存在領域を受信すると仮定すると、存在領域に対し、ノード i が受信したノードの存在領域の情報を記憶するのに必要となるメモリ空間は高々

$$(n+1) \frac{\pi \left(\frac{dia_{max}}{2} + V_{max} \cdot \Delta t_{max} + r \right)^2}{w^2}$$

個のグリッド分である。これは、障害物地図に対し $\frac{FS}{w}$ 個のグリッド分に相当する。ただし FS は障害物地図における自由空間を表す。存在領域の各グリッドに対し、縦方向と横方向のオフセットとそのグリッドの最短経路の距離をメモリに記憶する必要がある。ここで、4 章での実験で用いたパラメータ値を用いて具体値を計算する。対象となる領域を $500m \times 500m$ 、グリッドの大きさは $2m \times 2m$ とする。すなわち対象領域全体は 250×250 個のグリッドを持ち、 $[0, 250)$ 個のオフセットを表すのに 1 byte 必要となる。また、 $V_{max} = 2m/s$ 、 $r = 10m$ とすると、最短距離を表すのに 1 byte あれば十分である。よって、存在領域の各グリッドに対し 3 byte 必要となり、 FS の各グリッドに対し、2 byte 必要となる。4.2 節の実験結果から、存在領域は高々 $200m^2$ であり、 $dia_{max} = 25m$ である。よって、 $dia_{max}/2 = 12.5m$ に設定する。また実験結果から $\Delta t_{max} = 2$ 秒、 $n = 1$ とする。つまり、各ノードは 3 byte 必要とするグリッドを高々 1.1 キロ個持ち、2 byte 必要とする FS 内のグリッドを 62.5 キロ個持つ。よって必要となるメモリは約 130 kbyte である。これはメガバイトのオーダーでメモリをもつ最近の携帯端末にとって十分現実的な数値であると言える。

一方、時間計算量に関しては、UPL は画像処理ライブラリで 사용되는ような複雑な計算を必要とせず、リスト検索のようなシンプルな計算の繰り返しのみが要求されるため、処理能力の低いデバイスでも十分動作可能である。今後、我々は UPL が小型の携帯端末で実行可能であることを示すために、MOTE 上での実装を計画している。

6 まとめ

本稿では、都市歩行者のための位置推定手法 UPL を提案した。都市部では、建物などの障害物による無線ネットワークの分断やノードの移動により、シードからの位置情報を頻繁に得ることができない。このため、UPL では、シードからの位置情報に加え、移動する隣接ノードからの位置情報を利用する。また、障害物情報を利用し、障害物に対する回り込みなどを考慮したノードの移動可能領域を正確に求める。今後は、小型の携帯端末でも動作することを示すため MOTE への実装を行う予定である。

参考文献

- [1] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proc. of MobiCom 2004*, pp. 45–57, 2004.
- [2] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proc. of MobiCom 2001*, pp. 166–179, 2001.
- [3] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a global coordinate system from local information on an ad hoc sensor network," in *Proc. of IPSN '03*, pp. 333–348, 2003.
- [4] S. Guha, R. Murty, and E. G. Sifer, "Sextant: a unified node and event localization framework using non-convex constraints," in *Proc. of MobiHoc 2005*, pp. 205–216, 2005.
- [5] Network Simulator *MobiREAL web page*. <http://www.mobireal.net>.
- [6] K. Konishi, K. Maeda, K. Sato, A. Yamasaki, H. Yamaguchi, K. Yasumoto, and T. Higashino, "MobiREAL simulator – evaluating MANET applications in real environments –," in *Proc. of IEEE MASCOTS 2005*, pp. 499–502, 2005.
- [7] K. Maeda, K. Sato, K. Konishi, A. Yamasaki, A. Uchiyama, H. Yamaguchi, K. Yasumoto, and T. Higashino, "Getting urban pedestrian flow from simple observation: Realistic mobility generation in wireless network simulation," in *Proc. of MSWiM 2005*, pp. 151–158, 2005.
- [8] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications & Mobile Computing (WCMC)*, vol. 2, no. 5, pp. 483–502, 2002.
- [9] J. Broch, D. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols", in *Proc. of MobiCom 1998*, pp. 85–97, 1998.