

SmartFinder：集約型自己組織化スマートデバイス位置推定方式のノード間メトリックを用いた拡張とその実装

北之馬 貴正¹ 新居 英志¹ 森 流星¹ 安達 直世² 滝沢 泰久²

概要：空港、駅、工場、商業施設、オフィス、病院などの多様な屋内施設での人の活動状況やモノの利用状況を把握する試みにおいて、スマートフォンや BLE デバイスなどのモバイルスマートデバイスの位置は重要な情報である。既存方式は測位インフラとして測位設備や特性マップが十分に用意されていることを前提とし、導入・維持に大きなコストを必要とするため、屋内施設への適用は非常に困難となる。そこで、我々はスマートデバイスのトポロジ情報と 3 定点のみから高精度な位置推定ができる集約型自己組織化スマートデバイス位置推定方式 SmartFinder を提案している。しかし、SmartFinder が有効に機能するためにはトポロジ情報から構成する仮想メッシュネットワークがマルチホップトポロジを構成する必要がある。本稿では、位置推定対象ネットワークのトポロジ制約を排除するため、ノード間メトリックを用いて 1 ホップを詳細化して扱う SmartFinder を提案し、ノード間メトリックとして BLE の RSSI を用いた実装手法を示す。

1. はじめに

空港、駅、工場、商業施設、オフィス、病院などの多様な屋内施設において、旧来から人の活動状況やモノの利用状況の把握のため、それらの位置情報には高いニーズがある。この高いニーズに基づき、屋内施設における人の位置を人が携帯するスマートフォンの位置、モノの位置をモノに添付された BLE デバイスの位置とし、それらのスマートデバイスを追尾する屋内位置推定方式の研究開発が進められている。

既存方式は、屋内施設内にスマートデバイスを捕捉する高密度な測位設備／特性マップを構築および維持する必要があり、これらの導入、保守および拡張において著しいコストを必要とする。そのため、屋内施設への適用は、測位インフラの導入・維持に大きなコストを必要とし、非常に困難となる。従って、屋内環境においては測位設備やマップに依存しない自律性・柔軟性の高いモバイルスマートデバイスの位置推定方式が求められている。

我々はスマートデバイスのトポロジ情報と 3 定点のみから高精度な位置推定ができる集約型自己組織化スマートデバイス位置推定方式 SmartFinder [1] [2] を提案している。SmartFinder は、各ノードが隣接ノード情報を取得し、そ

の情報をクラウド環境上に集約して構成した仮想メッシュネットワークに自己組織化マップを応用した位置推定アルゴリズム、SOL(Self-Organizing Localization) を適用する。近傍トポロジ情報のみで多数の無線ノードの位置推定が可能であり、アンカーノード 3 点で絶対位置推定が可能のためアンカーノードへの依存度が極めて低い。SmartFinder はシミュレーション評価において、近傍トポロジ情報のみから高い位置推定が可能であり、その有効性が確認されている。しかし、SmartFinder が有効に機能するためにはトポロジ情報から構成する仮想メッシュネットワークがマルチホップトポロジを構成する必要がある。本稿では、位置推定対象ネットワークのトポロジ制約を排除するため、ノード間メトリックを用いて 1 ホップを詳細化して扱う SmartFinder を提案し、ノード間メトリックとして BLE の受信信号強度 (Received Signal Strength Indicator: RSSI) を用いた実装手法を示す。

2. 関連研究

屋内施設でのモバイルスマートデバイスの位置推定方式において、使用するデバイスの観点から利用もしくは研究されている方式を分類し概説する。

2.1 搬送波を用いた方式

2.1.1 Range-Based 方式

Range-Based 方式は位置推定処理にノード間の距離情報を利用するため、モバイルスマートデバイスにノード間通

¹ 関西大学 理工学研究科
Graduate School of Engineering, Kansai University

² 関西大学 環境都市工学部
Faculty of Environmental and Urban Engineering, Kansai University

信機能の他にノード間距離を測定するデバイス（測距デバイス）を必要とする。ノード間距離の測距には、Time Difference Of Arrival (TDOA), Time Of Arrival (TOA) が利用されている。

TOA 方式は、送信側から受信側に信号が到着するまでの時間を測定し、伝送媒体の伝送速度からノード間の距離を計算する方式である。TOA 方式を利用した位置推定方式として Global Positioning System (GPS) [3] や Ultra Wide Band(UWB) を用いた方式 [4] がある。GPS は衛星との見通しが遮られるため屋内での使用ができない。UWB は非常に短いパルスを用いることでノード間距離を測ることにより高精度な位置推定が可能であるが、通信距離が短いので多数のアンカーノードを必要とする。

TDOA 方式は、異なる2つの伝送媒体を用いて通信を行い、それらの到着時間の差からノード間の距離を計算する方式である。TDOA 方式を利用した位置推定方式としては、Active Bat [5], Cricket [6], Ubisense [7] や Iterative Multilateration [8] がある。

Range-Based 方式はこれらの測距デバイスで得られたノード間距離を使用し、三辺測量を用いて位置推定を行う。TOA 方式や TDOA 方式を用いた方式は高精度な位置推定が可能であるが、モバイルスマートデバイスに付加的な測距デバイスを用いる必要があるため、モバイルスマートデバイスの位置推定には適さないと考える。さらに、各ノードは3つ以上のアンカーノードとの見通し内 (Line-Of-Sight) の通信を必要とするため、位置推定には相当数のアンカーノードを必要とする。すなわち、これらの方式は測位設備に強く依存する。

2.1.2 Range-Free 方式

Range-Free 方式は、付加的な測距デバイスを用いず位置を推定する。代表的な方式として Centroid 方式 [9] や APIT 方式 [10] 等があり、一般的なモバイルスマートデバイスで容易に利用できる。Centroid 方式は、各ノードが通信可能な複数のアンカーノードの位置情報を取得し、それらの重心を自己位置として推定する方式である。APIT 方式は、複数個のアンカーノードの組み合わせから作成可能な全ての三角形に対して、位置を推定するノードが外側にあるか内側にあるかを判定し位置を推定する方式である。Centroid と APIT の位置推定精度はアンカーノード数に依存して改善を図れるが、その絶対精度は低い。さらに、各ノードは3つ以上のアンカーノードとの通信を必要とするため、位置推定には相当数のアンカーノードを必要とする。実用システムの iBeacon [11] や IMES [12] もこの方式に分類できるが、電波強度を用いたアンカーノードとの近接から位置推定を行うため絶対精度は低く、また、位置推定にはモバイルスマートデバイスの移動領域全体にアンカーノードを配置する必要があるが、大規模屋内施設では膨大な数になる。すなわち、これらの方式も測位設備に強く依存する。

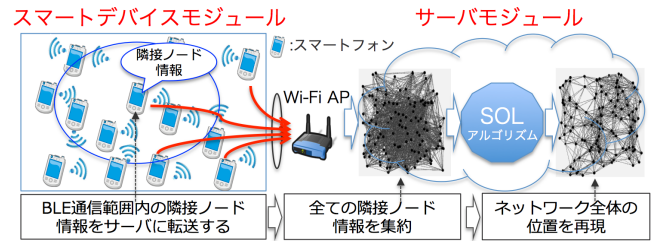


図 1 SmartFinder のシステム構成

2.2 センサを用いた方式

2.2.1 Pedestrian Dead Reckoning (PDR)

ジャイロセンサや加速度センサ等の各種モーションセンサを用いる Pedestrian Dead Reckoning (PDR) [13] は移動方向や移動距離を算出することで基準点からの相対位置を推定する方式である。そのため、絶対位置を得るには iBeacon や IMES 等と連携し基準点を推定する必要がある。さらに、移動における相対位置算出の誤差が累積するため、利用可能な精度を得るにはその精度補正のための基準点・補正点となるアンカーノードを移動空間全体に配置する必要がある。すなわち、PDR も測位設備を前提としてこれに依存する。

2.2.2 フィンガープリンティングを用いた方式

事前に施設内の電磁気や電波などの環境物理特性を計測して作成した特性マップとスマートデバイスが持つセンサの計測値を用いて、特性マップ上からそのスマートデバイスの位置を推定するフィンガープリンティングを用いた方式がある。地磁気を用いた方式 [14] や電波を用いた方式 [15] 等がある。これらの方式はアンカーノードが不要になるが、それに代わる施設内の特性マップが必要であり、これの作成のために環境物理特性の綿密な計測が事前に必要となる。すなわち、事前の特性マップ作成を必要としてこれに強く依存する。

3. SmartFinder

我々が提案した SmartFinder [1] [2] を概説する。

3.1 システム構成

一般的に、屋内施設におけるモバイルスマートデバイスに関するネットワーク環境は多数のモバイルスマートデバイスとこれらからの情報を Wi-Fi または LTE で集約するサーバから構成される。このような構成の無線ネットワークに着目し、図 1 に示すように、システムはスマートデバイスモジュールとサーバモジュールで構成する。モバイルスマートデバイスの位置を継続的に求めるため、以下の位置推定処理シーケンスを周期的に繰返し実施することで高精度なモバイルスマートデバイスの位置推定を実現する。

- 無線ノードモジュールによる隣接ノード ID の取得と転送
- サーバモジュールにおける無線ノードモジュールから

の隣接ノード ID リストの集約とデータロスによる隣接ノード情報の欠損を考慮した仮想メッシュトポロジ構成

- 大域/局所 SOL による人の移動速度に追従する仮想メッシュトポロジのノード位置推定

3.1.1 スマートデバイスモジュール

スマートデバイスはスマートフォンなどの Bluetooth Low Energy (BLE) と Wi-Fi/LTE の通信機能とモーションセンサを持つデバイスを想定する。スマートデバイスで動作するスマートデバイスモジュールは以下の処理を周期的に繰り返す。

- BLE を用いた自己 ID の広告ブロードキャストと広告ブロードキャスト受信による隣接ノードの ID 取得
- 自身の移動もしくは停止状態をモーションセンサを用いた判別 (移動/停止情報)
- Wi-Fi/LTE を用いたサーバへの隣接ノード ID リストと自身の移動/停止情報の送信

3.1.2 サーバモジュール

サーバで動作するサーバモジュールは以下のシーケンスを周期的に繰り返す、継続的にモバイルスマートデバイスの位置を推定する。

- 全てのモバイルスマートデバイスの隣接ノード ID リストと移動/停止情報を Wi-Fi または LTE を用いて集約し蓄積する。
- 集約した隣接ノード ID リストに基づき、仮想メッシュトポロジを構成/更新する。
- 仮想メッシュトポロジに集約型 SOL を適用することで全てのスマートデバイスの位置を推定する。

3.2 隣接ノード情報の欠損を考慮した仮想メッシュトポロジ構成と更新

隣接ノード情報の欠損は以下の通信時に発生する可能性があり、これらは位置推定精度の劣化要因となる。

- BLE を用いた隣接ノード ID の取得時
- Wi-Fi/LTE を用いたサーバへの隣接ノード ID リスト集約時

これらの欠損を補完するため、クラウドサーバモジュール上で一定期間保持した隣接ノード情報から双方向リンクの仮想メッシュトポロジを構成する。停止しているノード (停止ノード) 間のトポロジは変化しないため、隣接ノード情報の長期間の保持ができる。一方、移動しているノード (移動ノード) とその近傍ノードとのトポロジは変化するため、移動速度に追従した位置推定には直近の隣接ノード情報を必要とする。従って、停止しているノード (停止ノード) 間の隣接ノード情報保持期間 t^s は長い期間とし、移動ノードと他のノード間の隣接ノード情報保持期間 t^m は短い期間とする。以下のように、上記に期間に基づいて保持した隣接ノードリストから双方向リンクの仮想メッシュトポロジを構成する。

- ノード i の隣接ノード ID リストに含まれるノードと隣接ノード ID リストにノード i を含むノードをノード i の 1 次近傍ノードとする。
 - n 次近傍ノード x の隣接ノード情報に含まれるノード、または、隣接ノード情報としてノード x を含むノードにおいて、ノード i および $(n-1)$ 次までの近傍ノード群の 1 次近傍ノードに含まれないノードをノード i のノード x を中継する $(n+1)$ 次近傍ノードとする。
- 上記処理を全てのノードに実施し、個々のノード毎に多次近傍ノードを設定する。以上により、隣接ノード情報の欠損を考慮した仮想メッシュトポロジを構成し、精度劣化を抑制する。

3.3 モバイルスマートデバイスへ適用拡張した SOL アルゴリズム

移動ノードと停止ノードの位置推定戦略を分ける。停止ノードの位置推定は全体の仮想メッシュトポロジを用いて高精度に実施する (大域 SOL)。大域 SOL の計算時間のおおよその見積もりを数秒から数十秒程度と想定し、大域 SOL の実行周期は数十秒程度の長周期とする。移動ノードの位置推定は、局所的な仮想メッシュトポロジの範囲に限定し、かつ高精度な推定位置をもつ停止ノードのみを用いる。この位置推定アルゴリズムにより計算時間を短縮し、かつ精度維持を図る (局所 SOL)。局所 SOL の計算時間のおおよその見積もりを 1 秒未満と想定し、局所 SOL の実行周期は 1 秒程度の短周期とする。大域 SOL と局所 SOL の並列処理により移動ノードと停止ノードの位置推定を行う。以降、大域/局所 SOL 内での位置推定のための繰返し計算処理における 1 回あたりの処理単位をステップ、大域/局所 SOL 内の各ステップの計算過程位置を仮位置と定義する。隣接ノード ID リストの集約、仮想メッシュトポロジの構成、複数ステップから成る局所 SOL の繰返しシーケンスにおける 1 回あたりのシーケンス単位を 1 サイクル、大域/局所 SOL による推定の結果を位置推定結果と定義する。

3.3.1 大域 SOL

大域 SOL では十分な推定時間を利用できることから、次の位置推定処理サイクルを複数回実施して、最良の推定ジオメトリを位置推定結果として出力する。

- SOL アルゴリズムによる位置推定
- 絶対座標変換
- 推定ジオメトリの領域判定値算出
- 最小値の領域判定値と推定ジオメトリを記憶

SOL アルゴリズムは多次近傍ノードによる位置修正を繰返すことで相対ジオメトリを再現する。位置修正の初期段階は広い範囲の多次近傍ノードを用いて大域的なジオメトリを形成し、修正段階の進行に伴い位置修正に使用する多次近傍ノードのホップ数を減少させて局所的かつ詳細なジオメトリを形成し収束させる。従って、SOL アルゴリズム

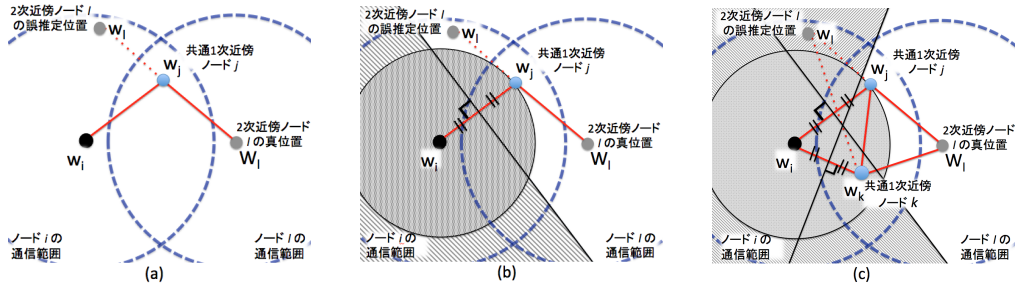


図 2 推定ノードのトポロジ矛盾領域

ムによる位置修正は以下のステップにより構成される。

[Step.1] 各ノードの推定位置をランダムに生成する。以降、 t 回目の修正におけるノード i の推定位置を $w_i(t)$ とする。

[Step.2] ノード i に対して H ホップとなるノード群からランダムにノード 1 つを選択し、これをノード h とする。ノード h を用いたノード i の修正ベクトル $V_i^{\{H\}}(t)$ において、ノード間距離をホップ数 H とし、次のように定義する。

$$V_i^{\{H\}}(t) = \frac{H - |w_i(t) - w_h(t)|}{|w_i(t) - w_h(t)|} (w_i(t) - w_h(t)) \quad (1)$$

修正ベクトル $V_i^{\{N\}}(t)$ を用い、ノード i の位置修正は次のように行う。

$$w_i(t+1) = \begin{cases} w_i(t) + \alpha_i(t) \cdot (V_i^{\{1\}}(t) + V_i^{\{H\}}(t)) & (t < \tau_H) \\ w_i(t) + \alpha_i(t) \cdot (V_i^{\{1\}}(t) + V_i^{\{H-1\}}(t)) & (\tau_H \leq t < \tau_{H-1}) \\ \vdots & \\ w_i(t) + \alpha_i(t) \cdot (V_i^{\{1\}}(t) + V_i^{\{2\}}(t)) & (\text{otherwise}) \end{cases} \quad (2)$$

$$\alpha_i(t) = \eta \alpha_i(t-1) \quad (0 < \eta < 1). \quad (3)$$

ただし、 τ_H は位置修正に用いる多次近傍ノードを切り替える修正回数の閾値、 $\alpha_i(t)$ はノード i の t 回目修正における学習係数である。

各ノードにおいて Step.2 を繰り返して位置修正を行い、ノード全体の相対ジオメトリを再現する。この相対ジオメトリを 3 点のアンカーノードの真位置と推定位置の対を用いて絶対座標へ変換し、各ノードの絶対位置を推定する。アンカーノードの真位置 $W_A = (X_A, Y_A)$ は推定位置 $w_A = (x_A, y_A)$ を用いて以下のように表される。

$$\begin{aligned} X_A &= ax_A + by_A + t_x \\ Y_A &= cx_A + dy_A + t_y \end{aligned} \quad (4)$$

3 つのアンカーノードから構成される連立方程式 (4) から 6 つの係数 a, b, t_x, c, d, t_y を得ることにより、すべてのノードは以下のように推定位置 $w_i = (x_i, y_i)$ から絶対座標 $\hat{w}_i = (\hat{x}_i, \hat{y}_i)$ へ変換し絶対推定位置を得る。

$$\begin{pmatrix} \hat{x}_i \\ \hat{y}_i \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \quad (5)$$

SOL アルゴリズムは位置修正に用いる近傍ノードをランダムに選択するため、同一のトポロジにおいても、位置推定誤差は変動する。そのため、トポロジ矛盾率を用いた推定ジオメトリ評価を行い、矛盾の少ないジオメトリを位置推定結果とする。図 2(a) にトポロジ矛盾の場合を示す。ノード i 、ノード i の 1 次近傍ノード j 、ノード i の 2 次近傍かつノード j の 1 次近傍ノード l のそれぞれの推定位置を w_i, w_j, w_l 、ノード l の真位置 W_l とすると、 w_l はトポロジ矛盾となる位置である。図 2(b) に示すように、基準点 w_i と w_j において、線分 $\overline{w_j - w_i}$ の垂直 2 等分線を用いて w_i と w_j のいずれかに近い領域に空間を 2 分割する (線分 $\overline{w_j - w_i}$ の垂直 2 等分線の左側が w_i に近い領域、右側が w_j に近い領域)。ノード l はノード i の 2 次近傍であるので、 w_l は w_j に近い領域内に位置しなければならない。従って、 w_i に近い領域にある ($|w_l - w_i| \leq |w_l - w_j|$) 場合トポロジ矛盾と判定する。さらに、図 2(c) に示すように、トポロジ矛盾の検知領域を拡大するため、上記と同様に全ての共通 1 次近傍群による複数の分割空間を重ね合わせてトポロジ矛盾の検知範囲を拡大し、誤推定検知の可能性を高める。共通 1 次近傍群による領域判定を行った回数におけるトポロジ矛盾の発生回数を領域判定値と定義する。領域判定値と位置推定誤差には、領域判定値が低下すれば位置推定誤差が小さくなるという一定の相関関係があるため、最小の領域判定値のジオメトリを位置推定推定結果とする。

3.3.2 局所 SOL

人の移動速度に追従可能な位置推定処理の時間制約において、局所 SOL は以下により計算時間の短縮と精度維持を両立する。

- 大域 SOL により高精度に推定された位置を持つ停止ノードの仮位置修正は実施せず、停止ノードを基準点として用いた移動ノードの仮位置修正を行う。
- 移動ノードの移動速度および更新周期から、局所 SOL 実行周期あたりの移動距離は局所的である。そのため、直近の位置推定結果から漸次的な局所更新とし、大域的な多次近傍ノードを用いた仮位置修正を削減する。

局所 SOL の学習係数は、大域 SOL と同様の収束を得るため、大域 SOL を基準とした局所 SOL の仮位置更新ステップ削減数に対応させて設定する。 S^g を大域 SOL に

おける仮位置修正回数, S^l を局所 SOL における仮位置修正回数とすると, 大域 SOL における仮位置更新ステップの終盤過程相当である ($S^g - S^l$) から S^g の学習係数を局所 SOL における学習係数として用いる. すなわち, 局所 SOL における学習係数の初期値 $\alpha_i(0)$ は式 (6) より算出する. ただし, t 回目の学習係数 $\alpha_i(t)$ は式 (3) に従う.

$$\alpha_i(0) = \eta \cdot \exp(S^g - S^l) \quad (0 < \eta < 1). \quad (6)$$

停止ノードの絶対推定位置を基準点とした仮位置更新を行う局所 SOL において, ホップ数を用いたノード間相対距離を絶対距離として用いるにはスケール調整を必要とする. 従って, 大域 SOL の絶対座標変換後の推定ジオメトリにおける各ホップ数 H の平均ノード間推定距離 $\bar{s}^{\{H\}}$ を以下により算出し, これをスケール調整に用いる.

$$\bar{s}^{\{H\}} = \frac{1}{|N||N^{\{H\}}|} \sum_{i=1}^{|N|} \sum_{h=1}^{|N^{\{H\}}|} d_{ih} \quad (7)$$

ただし, 絶対座標変換後の推定ジオメトリにおける全ノードの集合を N , その集合の要素数を $|N|$, ノード集合 N の任意のノード i に対して H ホップとなるノードを h , その集合を $N^{\{H\}}$ (ノード集合 N の部分集合), その集合の要素数を $|N^{\{H\}}|$ とし, ノード i とノード h の推定ノード間距離を d_{ih} と示す.

この局所 SOL による仮位置修正は以下のステップにより構成される.

[Step.1] 各ノードの直近の位置推定結果を各ノードの修正初期仮位置 $w_i(0)$ とする.

[Step.2] i が移動ノードであれば, i に対して 1 ホップとなる停止ノードと 2 ホップとなる停止ノードを 1 つずつ選択する. 修正ベクトル $V_i^{\{H\}}(t)$ は以下のように表される.

$$V_i^{\{H\}}(t) = \frac{H \cdot \bar{s}^{\{H\}} - |w_i(t) - w_h(t)|}{|w_i(t) - w_h(t)|} (w_i(t) - w_h(t)) \quad (8)$$

この修正ベクトル $V_i^{\{N\}}(t)$ を用いた局所 SOL における移動ノード i の位置修正は次のように行う.

$$w_i(t+1) = w_i(t) + \alpha_i(t) \cdot (V_i^{\{1\}}(t) + V_i^{\{2\}}(t)) \quad (9)$$

4. SmartFinder の拡張

SmartFinder はノード間相対距離を用いた仮位置修正の繰返しにより相対ジオメトリを再現し, アンカーノードを用いた座標変換により絶対位置を推定する. すなわち, ノード間の絶対距離を用いず相対距離のみで絶対位置推定が可能である. しかし, ホップ数をノード間相対距離として用いるため, マルチホップトポロジとなる仮想メッシュネットワークを必要とするが, 実際のネットワークは SmartFinder の位置推定アルゴリズムに適したトポロジを構成するとは限らない. この問題に対し, ノード間のメトリックを用いて 1 ホップを詳細化して扱うことにより, 任意のネットワークトポロジにおいても有効に機能する SmartFinder の拡張手法を提案する.

4.1 仮想メッシュネットワークの構成と更新

従来の SmartFinder は, 自然数のノード間メトリックを想定し, 仮想メッシュトポロジにおける最短ホップ数 (ノード間ホップ距離) をノード間相対距離として設定する. これを小数のノード間メトリックに対応する仮想メッシュネットワークに拡張する. そのため, ダイクストラ法を全ノードを始点として適用する方法等を用いて, ノード i とノード j 間のメトリックの最短経路をノード間メトリック距離 d_{ij} とし, これをノード間相対距離として用いる. 以上により, 小数のノード間メトリックを用いた仮想メッシュネットワークを構成する.

4.2 位置推定アルゴリズムの拡張

4.2.1 大域 SOL における拡張

複数回の位置推定処理サイクルから最良の推定ジオメトリを位置推定結果として出力する大域 SOL において, ノード間ホップ距離を用いた以下の処理をノード間メトリック距離に対応するように拡張する.

- SOL アルゴリズムによる位置推定
- 推定ジオメトリの領域判定値算出

一方, 以下の処理はノード間ホップ距離に依存しない処理のため従来の SmartFinder と同様の処理とする.

- 絶対座標変換
- 最小値の領域判定値と推定ジオメトリを記憶

SOL アルゴリズムによる位置推定において, 相対ジオメトリを再現するために多次近傍ノードによる位置修正を繰返す Step.2 の処理を拡張する. 位置更新においてノード間相対距離として用いるノード間ホップ距離をノード間メトリック距離 d_{ij} , ステップ数 t により決定される大域 SOL の位置修正に用いるノード集合を選択するためのノード間メトリック距離の上限閾値を $\gamma^g(t)$, ノード i に対してノード間メトリック距離が $\gamma^g(t)$ 以下となるノード群からランダムに 1 つ選択したノードをノード m とする. 位置修正の初期段階は大域的なジオメトリを形成し, 修正段階の進行に伴い局所的かつ詳細なジオメトリを形成し収束させるため, このノード m とのノード間メトリック距離 $d_{im}^{\{\gamma^g(t)\}}$ を用いたノード i の修正ベクトル $V_{im}^{\{\gamma^g(t)\}}(t)$ を, 次のように定義し,

$$V_{im}^{\{\gamma^g(t)\}}(t) = \frac{d_{im}^{\{\gamma^g(t)\}} - |w_i(t) - w_m(t)|}{|w_i(t) - w_m(t)|} (w_i(t) - w_m(t)) \quad (10)$$

この修正ベクトル $V_{im}^{\{\gamma^g(t)\}}(t)$ を用いたノード i の位置修正は次のように行う.

$$w_i(t+1) = w_i(t) + \alpha_i(t) \cdot (V_{im}^{\{1.0\}}(t) + V_{im}^{\{\gamma^g(t)\}}(t)) \quad (11)$$

ただし, この $\gamma^g(t)$ は式 (12) により決定する.

$$\gamma^g(t) = \begin{cases} d^{\max} - \frac{t(d^{\max} - 2.0)}{S^g} & (d^{\max} > 2.0) \\ 2.0 & (\text{otherwise}) \end{cases} \quad (12)$$

ただし, d^{\max} は全ノードにおけるノード間メトリック距離

の最大値とする。

矛盾の少ないジオメトリを推定するためのトポロジ矛盾率の算出による推定ジオメトリ評価は、メトリック距離矛盾率の算出による推定ジオメトリ評価として、以下のように拡張する。複数の1次数近傍ノードを用いた2次近傍ノードの矛盾判定において、1次数近傍ノードと2次近傍ノードの選択方法を変更することにより、ノード間メトリック距離に対応させる。ノード i において、ランダムに選択したノードを2次近傍ノード相当のノード l とし、ノード i とノード l に対するノード間メトリック距離がノード i とノード l 間のメトリック距離未満となるノード、すなわち、 $(d_{ij} \leq d_{il})$ かつ $(d_{jl} \leq d_{il})$ のノードを1次近傍ノード相当のノード j とする。従来の SmartFinder におけるトポロジ矛盾判定と同様に、 w_i と w_j を基準点とし、線分 $\overline{w_j - w_i}$ の垂直二等分線を用いて w_i と w_j のいずれかに近い領域に空間を2分割し、 w_i に近い領域にある $(|w_l - w_i| \leq |w_l - w_j|)$ 場合、メトリック距離矛盾と判定する。これにより、任意のノード間メトリック距離に対して相対距離の矛盾を検知することが可能となる。さらに、検知領域を拡大するため、上記と同様に全ての共通1次近傍相当のノード群による複数の分割空間を重ね合わせてメトリック距離矛盾の検知範囲を拡大し、誤推定検知の可能性を高める。全ての2次近傍ノードとの組み合わせを領域判定値とし、これにより最小の領域判定値のジオメトリを選出し、位置推定推定結果とする。

4.2.2 局所 SOL における拡張

大域 SOL で推定された停止ノードの位置を基準点とし、位置推定を行う局所 SOL において、ノード間ホップ距離を用いた以下の処理をノード間メトリック距離に対応するように拡張する。

- ノード間相対距離のスケール調整
- SOL アルゴリズムによる位置推定

一方、以下の処理はノード間メトリック距離に依存しない処理のため従来の SmartFinder と同様の処理とする。

- 局所 SOL で用いる学習係数の算出

ノード間メトリック距離をノード間の絶対距離として用いるには従来の SmartFinder と同様にスケール調整を必要とする。そのため、大域 SOL の座標変換前のネットワークスケールと座標変換後のネットワークスケールの比を用いる。すなわち、局所 SOL で用いるノード間メトリック距離は、大域 SOL での絶対座標変換前の推定ネットワーク形状に対する絶対座標変換後の推定ネットワーク形状のスケール比として求める。この、拡張した SmartFinder におけるスケール調整は \bar{s} を以下により算出し、これをスケール調整に用いる。

$$\bar{s} = \frac{1}{|N|C_2} \sum_{i=1}^{|N|-1} \sum_{j=i+1}^{|N|} \frac{|\hat{w}_i - \hat{w}_j|}{|w_i - w_j|} \quad (13)$$

局所 SOL の Step.2 による仮位置修正をノード間相対距

離に対応するように以下のように拡張する。ノード i に対してノード間メトリック距離が 1.0 もしくは 2.0 の閾値 γ^l 以下となるノード群からランダムにノードを1つ選択し、これをノード m とする。このノード m とのノード間メトリック距離 $d_{im}^{\{\gamma^l\}}$ を用いたノード i の修正ベクトル $V_{im}^{\{\gamma^l\}}(t)$ を、次のように定義する。

$$V_{im}^{\{\gamma^l\}}(t) = \frac{d_{im}^{\{\gamma^l\}} \cdot \bar{s} - |w_i(t) - w_m(t)|}{|w_i(t) - w_m(t)|} (w_i(t) - w_m(t)) \quad (14)$$

この修正ベクトル $V_{ij}^{\{\gamma^l\}}(t)$ を用いた局所 SOL における移動ノード i の位置修正は次のように行う。

$$w_i(t+1) = w_i(t) + \alpha_i(t) \cdot (V_{im}^{\{1.0\}}(t) + V_{im}^{\{2.0\}}(t)) \quad (15)$$

5. 実装手法

隣接ノードとの距離に相関があるメトリックをスマートデバイスモジュールで取得し、これに基づき、サーバモジュールで1ホップを詳細化した仮想メッシュネットワークを構成する。SmartFinder の位置推定アルゴリズムはおおよその隣接ノードとの距離相関でも位置推定が可能であるため、本稿では、隣接ノードとのメトリックとして RSSI を用いる。RSSI は BLE の標準プロトコルで取得可能であるため、システム構成は従来の SmartFinder と同様とする。

5.1 実装システム構成

実装において、スマートデバイスには Android スマートフォンである FREE TEL の SAMURAI REI (1.3GHz 64bit, RAM 2GB), Wi-Fi アクセスポイント (Wi-Fi AP) には BUFFALO の WAPM-1750D, DHCP サーバには Raspberry Pi 3 Model B, SmartFinder サーバには MacBook Pro (CPU 3.3GHz, RAM 16 GB) を用いる。また、スマートフォン間の隣接ノード情報取得には BLE を用い、隣接ノード情報集約のための Wi-Fi AP との通信には BLE との電波干渉を予め避けるため IEEE 802.11a を用いる。スイッチを介して、スマートフォンを収容する Wi-Fi AP と DHCP サーバ, SmartFinder サーバをスター型に有線接続する。

5.1.1 スマートデバイスモジュール

スマートデバイスモジュールにおけるノード間 RSSI の取得は、BLE を用いた広告ブロードキャスト受信による隣接ノード ID 取得時に行う。これにより取得した各々の隣接ノード ID に対応する RSSI を隣接ノード ID リストに付加し、Wi-Fi/LTE を用いたサーバへの隣接ノード ID リストの送信により、隣接ノード ID に対応する RSSI をサーバに転送する。

5.1.2 サーバモジュール

RSSI は BLE による隣接ノード ID 取得の広告ブロードキャストが欠損すると計測不可となるため欠損する。ま

た、隣接ノード ID リストに RSSI を付加するため、隣接ノード ID リストが Wi-Fi/LTE を用いたサーバへの収集時に欠損すると RSSI も欠損する。これらの欠損を補完するため、サーバモジュールは隣接ノード ID に加え、隣接ノード ID に対する RSSI を保持する。これに基づき仮想メッシュネットワークを構成/更新し、位置推定を行う。

5.2 RSSI を用いたホップ数の詳細化

ノード間の RSSI に基づいてノード間の相対距離を算出する。フェージングによる RSSI の瞬時変動の影響を低減させるため、ノード i が受信したノード j の受信電力とノード j が受信したノード i の受信電力の隣接ノード保持期間における平均受信電力 \bar{P}_{ij} を相対距離算出に用いる。ここで、伝搬損失係数を ν とすると、平均受信電力は距離の ν 乗に反比例して減衰し、 ν は理想空間では 2、シャドウイングやマルチパスフェージングの影響下では 2 以上となることが知られている。SmartFinder ではノード間メトリックの最短経路をノード間相対距離とすることから、少なくとも距離に対して単調増加となるノード間メトリックであれば良い。そのため、本稿では基本評価として $\nu = 2$ を用いる。さらに、全てのノード間における最小受信電力 P_{\min} を用いて $\bar{P}_{ij}^{-\frac{1}{\nu}}$ を正規化して \bar{h}_{ij} とし (式 (16))、これをノード間メトリックとして扱う。

$$\bar{h}_{ij} = \left(\frac{\bar{P}_{ij}}{P_{\min}} \right)^{-\frac{1}{\nu}} \quad (16)$$

この \bar{h}_{ij} は 1 ホップノードに対する 0 以上 1 以下の相対距離を示す。すなわち、 \bar{h}_{ij} により、1 ホップノードとの相対距離を詳細化することが可能である。

6. 評価

6.1 評価方法

実験諸元を表 1 に、実験風景を図 3 に示す。SmartFinder のパラメータは表 2 に示す。ただし、隣接ノード情報の更新周期および局所 SOL 実行周期である 1 サイクルは 1 秒と想定する。停止ノードの真位置は縦 20 × 横 18 のグリッドから構成される計 360 の格子点からランダムに選択する。ただし、同一の格子点に複数のノードは配置せず、アンカーノードは実験諸元に示す座標の格子点に優先的に配置する。

6.2 評価方法

推定位置精度の評価として絶対位置評価を行う。絶対位置評価は、推定された各ノードの位置と真位置のユークリッド距離の平均である位置推定誤差 ERR_{ave} を用いて評価する。 ERR_{ave} は次の式 (17) のように求める。 s はサイクル、 $W_i(s)$ は s サイクル目のノード i の真位置、 $w_i(s)$ は推定位置を示す。

表 1 実験諸元

フィールド範囲 (m × m)	25 × 20
全ノード数	50
各試行回数	5
アンカーノード数	3
アンカーノード座標 (m)	(8.79, 14.13), (5.12, 5.53), (12.51, 5.52)
移動ノード数	1
1 サイクルあたりの移動量 (m/sec)	1

表 2 SmartFinder のパラメータ

減衰定数 η	0.992
大域 SOL の実行周期 (サイクル)	10
局所 SOL の実行周期 (サイクル)	1
大域 SOL における仮位置修正回数 S^g	800
局所 SOL における仮位置修正回数 S^l	400
停止ノード間の隣接ノード情報保持期間 t^s (サイクル)	60
移動ノードと他のノード間の隣接ノード情報保持期間 t^m (サイクル)	1

$$ERR_{\text{ave}} = \frac{1}{|N|} \sum_{i=1}^N |W_i(s) - w_i(s)| \quad (17)$$

これにより、個々のノード位置が絶対位置として正しく推定されているかを評価する。

6.3 評価結果

SmartFinder の位置推定において、10 サイクルまでは大域 SOL は未実施であるため、停止ノードの位置はランダムな位置となる。従って、局所 SOL ではランダムな位置である停止ノードの位置を基準点として用いるため、移動ノードの位置推定結果もランダム相当となる。この点を考慮して、絶対位置推定評価において、その対象とする時間は 10 サイクルから 180 サイクルとする。

図 4 に各時間における停止ノードと移動ノードの絶対位置推定評価を示す。これにより、大域 SOL により推定された停止ノードの位置推定精度と局所 SOL により推定された移動ノードの位置推定精度の時間遷移比較を行う。10 サイクル以降は大域 SOL と局所 SOL の位置推定結果が反映されるため高精度な位置推定結果を得る。10 サイクル以降の大域 SOL の位置推定誤差は 10 ステップ毎に停止ノードの位置推定結果を更新するため変動する。一方、局所 SOL はステップ毎に推定位置を出力するため、位置推定誤差はステップ毎に変動する。移動ノードは隣接ノード情報を長時間蓄積できないため、大域 SOL よりも電波環境の影響を受けていると考えられる。しかし、10 サイクルから 180 サイクルの ERR_{ave} の平均は、停止ノード約 1.637(m)、移動ノード約 2.457(m) であり、その精度差は単位時間あたりの移動量約 1(m) 以内に留まる。スマートデバイスモジュールにおける広告ブロードキャスト受信時から 1 秒間



図 3 実験風景

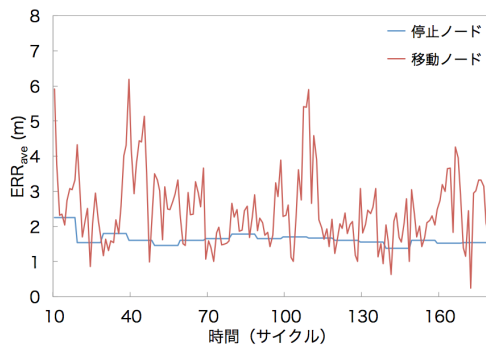


図 4 停止ノードと移動ノードの10サイクルから180サイクル間の絶対位置推定評価の推移

隣の隣接ノード ID リスト送信時までの送信待機遅延期待値と、サーバモジュールにおける隣接ノード ID リスト受信時から1秒間隔の局所 SOL 実行時までの実行待機遅延期待値の合計は1秒程度となる。局所 SOL の所要時間は微小なため、実行待機遅延期待値の合計はおおよそ位置推定遅延時間に相当する。すなわち、大域 SOL と局所 SOL の精度の差は隣接ノード集約遅延時間の期待値における移動量以内に留まる。そのため、ノード間メトリックを用いて1ホップを詳細化する手法においても大域 SOL と局所 SOL に分けた位置推定戦略は有効であると考えられる。

7. まとめ

本稿では、位置推定対象ネットワークのトポロジ制約を排除するため、ノード間メトリックを用いて1ホップを詳細化して扱う SmartFinder を提案し、ノード間メトリックとして BLE の RSSI を用いた実装手法を示した。さらに、実装評価から以下の有効性を確認した。

- ノード間 RSSI を用いて1ホップを詳細化したノード間メトリック距離を用いた SmartFinder により、アンカーノード3点のみで高精度な位置推定が可能である。
- ノード間メトリック距離を用いた SmartFinder においても、大域 SOL と局所 SOL に分けた位置推定戦略は有効に機能し、人の移動速度に追従した1秒毎の位置推定が可能である。

参考文献

- [1] Kitanouma, T., Nii, E., Takashima, Y., Adachi, N. and Takizawa, Y.: SmartFinder: Cloud-based self organizing localization for mobile smart devices in large-scale indoor facility, 2017 Global Internet of Things Summit (GIoTS), pp.1–6 (2017).
- [2] 北之馬貴正, 新居英志, 安達直世, 滝沢泰久: SmartFinder: 大規模屋内施設における集約型自己組織化スマートデバイス位置推定方式とその評価, 情報処理学会論文誌, Vol.59, No.2, pp.462–472 (2018).
- [3] Hofmann-Wellenhof, B., Lichtenegger, H. and Collins, J.: Global Positioning System, Theory and Practice, 4th ed. (1997).
- [4] Mikhaylov, K., Petäjälä, J., Hämmäläinen, M., Tikanmäki, A., and Kohno, R.: Impact of IEEE 802.15.4 Communication Settings on Performance in Asynchronous Two Way UWB Ranging, International Journal of Wireless Information Networks, pp.1–16 (2017).
- [5] Harter, A., Hopper, A., Steggle, P., Ward, A. and Webster, P.: The anatomy of a context-aware mobile applications, *Proc. ACM/IEEE MobiCom 99*, Vol.8, pp.187–197 (1999).
- [6] Priyantha, N., Miu, A., Balakrishnan, H. and Teller, S.: The cricket compass for context-aware mobile applications, *Proc. MOBICOM 2001* (2001).
- [7] Wozniak, M., Odziemczyk, W. and Nagorski, K.: Investigation of Practical and Theoretical Accuracy of Wireless Indoor Positioning System Ubisense, *Reports on Geodesy and Geoinformatics*, Vol. 95, No.1, pp.36–48 (2013).
- [8] Savvides, A., Han, C. and Srivastava, M.: Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors, *Proc. ACM MobiCom 2001*, pp.1–14 (2001).
- [9] Bulusu, N., Heidemann, J. and Estrin, D.: GPS-less low cost outdoor localization for very small devices, *IEEE Pers. Commun.*, Vol.7, No.5, pp.28–34 (2000).
- [10] He, C., Huang, C., M.Blum, B., A.Stankovic, J. and F.Abdelzaher, T.: Range-free localization and its impact on large scale sensor networks, *ACM TECS*, Vol.4, No.4, pp.877–906 (2005).
- [11] Nic, N.: Apple iBeacon technology briefing, *Journal of Direct, Data and Digital Marketing Practice* 15.3, pp.222–225 (2014).
- [12] 石井真, 小暮聡, 神武直彦, 海老沼拓史: IMES (Indoor Messaging System) の原理と課題及びその解決について, *GPS/GNSS Symposium 2009 テキスト*, pp.120–125 (2009).
- [13] Li, F., Zhao, C., Ding, G., Gong, J., Liu, C. and Zhao, F.: A reliable and accurate indoor localization method using phone inertial sensors, *Proc. the 2012 ACM Conference on Ubiquitous Computing (UbiComp '12)*, pp.421–430 (2012).
- [14] Vandermeulen, D., Vercauteren, C. and Weyn, M.: Indoor localization using a magnetic flux density map of a building, In *The Third International Conference on Ambient Computing, Applications, Services and Technologies*, pp.42–49 (2013).
- [15] Kawauchi, K. and Rekimoto, J.: FineMesh: High-Density Sampling Platform Using an Autonomous Robot, *Green Computing and Communications (Green-Com)*, 2012 IEEE International Conference on, *Besancon*, pp.477–486 (2012).