

# 3次元環境における無線センサネットワークの 集約型自己組織化ノード位置推定方式とその精度評価

毛利 友紀<sup>1</sup> 北之馬 貴正<sup>1</sup> 安達 直世<sup>2</sup> 滝沢 泰久<sup>2</sup>

**概要:** 無線センサネットワークにおいて、センシングデータの取得位置は重要な情報である。そこで我々は、各ノードから得た近傍ノード情報を元にクラウド上で自己組織化マップ (SOM) を用いて位置推定を行う、集約型自己組織化ノード位置推定方式 (集約型 SOL (Self-Organizing Localization)) を提案している。集約型 SOL は極少数のアンカーノードを使用し、測距デバイスを用いずに、高精度な位置推定が可能であり、クラウド上で位置推定を行うため各ノードへの負荷も軽減される。現在我々は2次元環境において高精度な位置推定を実現出来ているが、都市工学への適用を想定した場合、3次元の位置情報が必須となる。以上のことから、本稿では3次元の位置推定に対応した集約型自己組織化ノード位置推定方式を提案し、その精度評価から有用性を示す。

**キーワード:** 無線センサネットワーク, 位置推定, 3次元

## 1. はじめに

無線センサネットワーク (WSN) は, Internet of Things[1] や, Machine to Machine[2] などの物理情報処理をインターネットに取り込む試みにおいて, 必須技術であり, そのセンサノードの位置は重要な情報である。物理情報を利用したサービスとしては, 広範囲に大量のセンシングデバイスを散布することを想定した, 環境モニタリング, 構造物ヘルスマニタリング, 災害対策等が考えられている。

現在, 利用もしくは研究されている無線ノード位置推定方式は, ノード間測距デバイスの使用の観点から分類すると, Range-Based と Range-Free に大別できる。前者は測距デバイスを用いることにより高精度な位置推定が可能であるが, 特別なデバイスを必要とするためコストが高くなり無線センサノードに不適である。後者は測距デバイスが不要であるが, 高精度な位置推定をするためには, 十分な数のアンカーノードにより構成された空間を必要とし, 任意の広範囲な空間に適用することは困難である。

上記問題を解決するため, 我々は自己組織化マップを用いたノード位置推定方式 (SOL: Self-Organizing Localization) [3][4] をクラウド上で適用する集約型 SOL を提案した。集約型 SOL は, 各ノードが近傍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無線セ

ンサネットワーク (仮想 WSN) に SOL を適用して位置推定を行う。また近傍ノード情報のみで位置推定可能であるためアンカーノードへの依存度が極めて低く, クラウド上では多ホップの近傍ノード情報が利用できるため高精度な位置推定が可能である。さらに各ノードは近傍ノード情報を収集するだけでよいため, 低消費電力が望まれる WSN において特に有効な位置推定であると言える。

現在我々はこの集約型 SOL を用いて2次元において他方式よりも高精度な位置推定を実現しているが, 実際に構造物ヘルスマニタリングなどに適用する場合, 2次元の位置情報だけでは不十分であり, 3次元の位置推定が必須となる。

そこで本論文では3次元に適応した集約型 SOL を提案し, その精度評価を行う。

## 2. 関連研究

現在, 利用もしくは研究されている無線ノードの位置推定方式は, ノード間測距デバイスの使用の観点から分類すると, Range-Based と Range-Free に分類できる。

### 2.1 Range-Based

Range-Based 位置推定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, センサノードにノード間通信機能の他にノード間距離を測定するデバイス (以降, 測距デバイス) を必要とする。ノード間距離の測距には, TDOA (Time

<sup>1</sup> 関西大学大学院 理工学研究科

<sup>2</sup> 関西大学 環境都市工学部

Difference Of Arrival), TOA(Time Of Arrival) 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最も一般的なものは GPS である。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 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。しかし, TOA 方式や TDOA 方式を用いた位置推定技術は精度が高いが, センサノードに特別な測距デバイスを用いる必要があり, センサノードの消費電力やコストの面において WSN での利用は不向きであると考えられる。また現状では GPS 以外どの方式も 3 次元には対応出来ていない。

## 2.2 Range-Free

RangeFree 位置推定方式は, 位置推定に測距デバイスを用いない方式である。Centroid 方式 [9], APIT 方式 [10] や DV-Hop 方式 [11][12] 等がある。Centroid 方式は, 通信可能な複数のアンカーノードの位置情報を位置推定を行うノードが取得し, それらの重心を利用することで自身の位置を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, 位置推定を行うノードが外側にあるか内側にあるかを判定することで自身の位置を推定する方式である。DV-Hop 方式は, 3 つ以上のアンカーノードからのホップ数とアンカーノード間の距離から算出された 1 ホップの平均距離を利用して, アンカーノードとの距離を見積もることにより自身の位置を推定する方式である。これらの方式は少なくとも 3 つ以上のアンカーノードが必要であり, 精度の向上には多量なアンカーノードが必要なため広範囲な空間への適用には十分な事前準備が必要である。そのため, 適用可能な環境は限定的となる。また RangeFree 位置推定方式の場合, 現在 3 次元に対応出来ている方式は存在しない。

## 3. 集約型 SOL

集約型 SOL はノード間通信による電力消費の増大という問題を解決するため, クラウドコンピューティングを利用している。集約型 SOL は, 各センサノードからの隣接ノード情報をクラウド環境へ集約し, これにより構成する仮想 WSN へ SOL を適用した方式である。

### 3.1 位置推定のアルゴリズム

SOL は, 3 つのステップにより位置推定を行う。

[Step.1] 各ノードは, 自己位置をランダムに生成し,

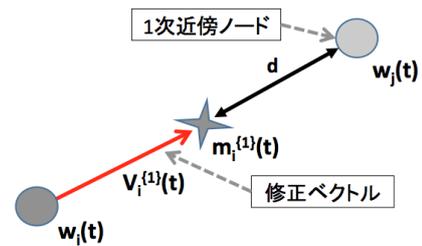


図 1 1 次近傍ノードによる位置修正

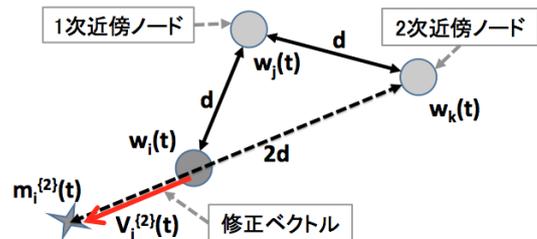


図 2 2 次近傍ノードによる位置修正

これを仮自己位置とする。以降, ノード  $i$  の修正  $t$  回目の仮自己位置を  $w_i(t)$  と表記する。各ノードはこの仮自己位置を 1 次近傍 (1 ホップ) ノードにブロードキャスト送信する。

[Step.2] 1 次近傍ノード  $j$  から仮位置情報  $w_j(t)$  を受信したノード  $i$  は, 仮自己位置  $w_i(t)$  と 1 次近傍ノード  $j$  の仮位置  $w_j(t)$  の直線上かつノード  $j$  から 1 ホップ距離  $d$  である位置を入力ベクトル  $m_i^{\{1\}}(t)$  とする。1 ホップ距離  $d$  とは, 測距デバイスを用いないために用いる相対的なノード間距離である。その後, ノード  $i$  の仮自己位置と入力ベクトル  $m_i^{\{1\}}(t)$  の距離  $|m_i^{\{1\}}(t) - w_i(t)|$  が最小となるような修正ベクトル  $V_i^{\{1\}}(t)$  を生成することにより, ノード  $i$  の仮自己位置を入力ベクトル  $m_i(t)$  に近づける (図 1)。

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

また, 1 次近傍ノード  $j$  の 1 次近傍ノード集合のうち, ノード  $i$  から 2 ホップにあたるノード (以降, 2 次近傍ノード)  $k$  の仮位置とホップ数  $2d = 2$  により推定される位置を入力ベクトル  $m_i^{\{2\}}(t)$  とする。ここでの入力ベクトルは, ノード  $i$  と 2 次近傍ノード  $k$  とのホップ数  $2d$  として生成する。その後, ノード  $i$  の仮位置をこの入力ベクトル  $m_i^{\{2\}}(t)$  に近づけるため, 次のような修正ベクトル  $V_i^{\{2\}}(t)$  を生成する (図 2)。

$$V_i^{\{2\}}(t) = \frac{d + d - |w_i(t) - w_k(t)|}{|w_i(t) - w_k(t)|} (w_i(t) - w_k(t)) \quad (2)$$

この 2 つの修正ベクトルを用いて次のように仮自己位置を修正する。

- ノード  $i$  の仮自己位置が 1 次近傍ノード  $j$  より 2 次近傍ノード  $k$  から遠い場合, 2 次近傍ノードとの相対関係に誤りがないことから 1 次近傍修正ベクトルのみで修正する。

- ノード  $i$  の仮自己位置が 1 次近傍ノード  $j$  より 2 次近傍ノード  $k$  に近い ( $|w_i(t) - w_j(t)| \geq |w_i(t) - w_k(t)|$ ) 場合 (トポロジ矛盾 (図 3)), 1 次近傍ノードとの相対関係に誤りがあるため, 1 次近傍修正ベクトルと 2 次近傍修正ベクトルの両方から仮自己位置  $w_i(t)$  を修正する.

以上の修正を定式化すると次のようになる.

$$w_i(t+1) = \begin{cases} w_i(t) + \alpha_i(t) \cdot V_i^{\{1\}}(t) & (|w_i(t) - w_j(t)| < |w_i(t) - w_k(t)|) \\ w_i(t) + \alpha_i(t) \cdot (V_i^{\{1\}}(t) + V_i^{\{2\}}(t)) & (|w_i(t) - w_j(t)| \geq |w_i(t) - w_k(t)|) \end{cases} \quad (3)$$

$\alpha_i(t)$  は  $t$  回目の修正時のノード  $i$  の学習関数であり, 次のようになる.

$$\alpha_i(t) = \eta \alpha_i(t-1) \quad (0 < \eta < 1) \quad (4)$$

ただし,  $\eta$  は正の減衰定数である.

[Step.3] 自身の仮自己位置  $w_i(t)$  に加えて, ランダムに 1 次近傍ノードの仮自己位置  $w_j(t)$  を選択し, これら 2 つのノードの仮自己位置を近傍ノードへブロードキャストする.

以上の Step.2 および Step.3 を繰り返す, 各ノードは自己位置を推定し, ネットワークトポロジを再現する.

### 3.2 位置推定補正処理

SOL ではノード間は距離はホップ数を用いているため, ノード推定位置は多くの誤差を含んでいる可能性がある. また, ノード推定位置はネットワーク内での相対位置である. 従って, 次のノード推定位置の補正処理を行う.

- 推定トポロジにおけるトポロジ矛盾の判定
- トポロジ矛盾を抑制するノード間修正距離による推定再試行
- 推定トポロジを絶対座標へ変換

#### 3.2.1 推定トポロジにおけるトポロジ矛盾の判定

位置推定処理が収束状態 (学習関数  $\alpha_i(t)$  が一定の閾値以下) になった段階でトポロジの矛盾の判定を開始する. トポロジ矛盾判定は各ノードにおいて推定位置が 1 次近傍ノードより 2 次近傍ノードに近い場合 (図 3) をトポロジ矛盾とする. この判定を複数の 1 次近傍ノードと 2 次近傍ノードで実施し, 次式を満たさない場合, 当該ノードの推定位置はトポロジ矛盾ありと判定する.

$$\frac{I_i^{\{2\}}}{N_i^{\{2\}}} < \lambda \quad (5)$$

$I_i^{\{2\}}$  は, ノード  $i$  におけるトポロジ矛盾と判定した 2 次近傍ノード数,  $N_i^{\{2\}}$  は, 判定に用いた 2 次近傍ノード数,  $\lambda$  は, トポロジ矛盾閾値である.

トポロジの矛盾が発生していると判定された場合, 全

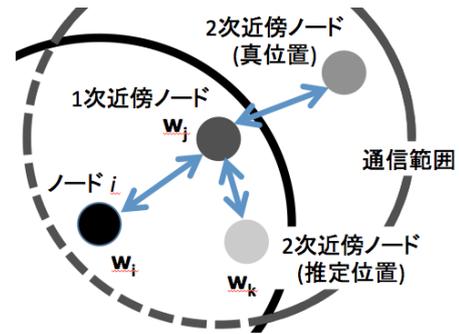


図 3 トポロジ矛盾

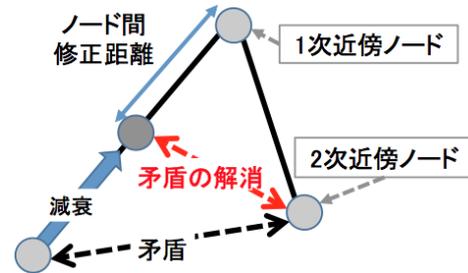


図 4 ノード間トポロジ矛盾を解消するノード間修正距離

ノードに対して位置推定処理の再試行を通知するメッセージを送信する. メッセージを受信した 1 次近傍ノードは, 自身の学習関数  $\alpha_i(t)$  を初期値の 1 に戻し, 位置推定処理を再試行する.

#### 3.2.2 ノード間トポロジ矛盾を解消するノード間修正距離

SOL はホップ数をノード間距離として用いる. しかし, ノード間距離をホップ数 1 とする入力はノード間距離を均等化するトポロジ再現へ制御するため, トポロジ矛盾の要因となる. 特に, 実トポロジにおいて 1 次近傍ノードと 2 次近傍ノードの距離より該当ノードと 1 次近傍ノードの距離が小さく距離の差が大きい場合にトポロジ矛盾の可能性が高くなる. 従って, トポロジ矛盾の場合, 該当ノードが上記のトポロジであると仮定し, 該当ノードにおける 1 次近傍ノードとの距離をホップ数 1 より小さい値 (正の実数値) に修正する (図 4). これをノード間修正距離と呼び, 式 (6) のように算出して位置推定処理を再試行する.  $T$  は再試行回数であり,  $d_T$  は再試行時のノード間修正距離 (初期値  $d_0$  は 1) である.

$$d_{T+1} = \frac{d_T}{T+1} \quad (6)$$

### 3.3 位置推定処理によるノード間通信の削除

WSN は 1 つのシンクノードと多数のセンサーノードから構成される. 各センサーノードは自身の隣接ノードへ広告として自身の ID をデータとするブロードキャストを行う. この広告ブロードキャストにより各ノードは隣接ノード ID を取得する. 各センサーノードは取得した隣接ノード ID を隣接ノード情報としてシンクノードへ転送し, シンクノードはこれらの情報をクラウド環境へ転送する. 集約

型 SOL はその転送された隣接ノード情報により構成する仮想 WSN へ SOL を適用することで、それぞれのノードは、SOL を実行する必要がなくなり、各センサノードの通信は広告ブロードキャストと隣接ノード情報のシンクノードへの転送のみとなる。従って、SOL の位置推定処理はクラウド環境で構成される仮想 WSN で実施されるため、各センサノードは SOL の位置推定処理に伴うノード間通信は一切必要としない。

### 3.4 SOL のクラウド環境への適用

仮想 WSN はすべてのノードの隣接ノード情報を集約するため、完全なトポロジ情報を有する。一方、そのトポロジのジオメトリ (形状) は全くのランダムである。集約型 SOL は、この完全なトポロジ情報をもつ WSN に SOL を適用し、推定トポロジの折れ曲がりを防ぐため以下の処理を行う。

- 多ホップノードによる位置修正
- 共通 1 次近傍群領域判定によるトポロジ矛盾検知  
なお共通 1 次近傍群領域判定によるトポロジ矛盾検知については第 4 章の 3 次元集約型 SOL の章にて説明を行う。

#### 3.4.1 仮想 WSN の構成

各センサノードは次のように近傍ノード情報を収集し、クラウド環境で仮想的なネットワーク (仮想 WSN) を作成する。

- 自己ノード ID をデータとしてブロードキャスト (広告ブロードキャスト) 送信する。
- 各センサノードは受信した広告ブロードキャストから隣接ノードの ID を取得し、取得したノード ID リストを隣接ノード情報としてシンクノードに転送する。
- シンクノードはこれをクラウド環境へ転送する。

以上により、クラウド環境は全てノードの隣接ノード情報を取得し、保持する。集約型 SOL は上記の取得された隣接ノードを用いてクラウド上で次のように近傍トポロジを構成する。

- ノード  $i$  の隣接ノード情報に含まれるノードをノード  $i$  の 1 次近傍ノードとする。
- 上記 1 次近傍ノード  $j$  の隣接ノード情報に含まれるノードで、ノード  $i$  およびノード  $i$  の隣接ノード情報に含まれないノードをノード  $i$  のノード  $j$  を中継する 2 次近傍ノードとする。
- 同様に、 $n$  次近傍ノード  $x$  の隣接ノード情報に含まれて、 $(n-1)$  次までの近傍ノード群の隣接ノード情報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。
- 上記処理を再帰的に繰り返し、ノード  $i$  の近傍トポロジを拡大し、ネットワーク全体を通してノード  $i$  の多次近傍ノードを設定する。

以上の処理をクラウド上で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し、これを仮想 WSN とする。

#### 3.4.2 多次近傍ノードによる位置更新

集約型 SOL は、3.2.1 節で述べたように、各ノード毎に多次近傍ノード情報を保持する。この多次近傍ノード情報を用いることにより、位置更新に用いる近傍ノードの範囲を大幅に拡大することが出来る。多次近傍ノード情報は、ホップ数に応じてノード間距離が増加するように構成されている。すなわち、選択されるノードは、ノード  $i$  から  $n$  次近傍ノード以下であり、かつノード  $i$  からの  $(n-1)$  次近傍ノードのいずれのノードよりもノード  $i$  から遠方に位置する条件を満たす。従って、一定距離  $d \times$  ホップ数  $n$  により位置更新を行う。多ホップによる  $n$  次近傍修正ベクトルは式 (7) から求める。 $w_n(t)$  は  $n$  次近傍ノードの仮自己位置である。さらに、位置修正の初期段階は広い範囲の近傍ノードを用いて大域的なトポロジを形成し、修正段階の進行に伴い、位置修正に使用する近傍ノードのホップ数を減少させて、局所的かつ詳細なトポロジを形成し、収束させる。従って、次式 (8) のように仮自己位置情報の更新を行う。 $\tau_N$  は推定の段階を分ける閾値である。

$$V_i^{\{n\}}(t) = \frac{nd - |w_i(t) - w_n(t)|}{|w_i(t) - w_n(t)|} (w_i(t) - w_n(t)) \quad (7)$$

$$w_i(t+1) = \begin{cases} w_i(t) + \alpha_i(t) \cdot (V_i^{\{1\}}(t) + V_i^{\{n\}}(t)) & (t < \tau_n) \\ w_i(t) + \alpha_i(t) \cdot (V_i^{\{1\}}(t) + V_i^{\{n-1\}}(t)) & (\tau_n < t < \tau_{n-1}) \\ \vdots & \\ w_i(t) + \alpha_i(t) \cdot (V_i^{\{1\}}(t) + V_i^{\{3\}}(t)) & (\tau_4 < t < \tau_3) \\ w_i(t) + \alpha_i(t) \cdot (V_i^{\{1\}}(t) + V_i^{\{2\}}(t)) & (\text{otherwise}) \end{cases} \quad (8)$$

## 4. 3 次元集約型 SOL

集約型 SOL の位置推定アルゴリズムや更新方法などは、ノードの位置関係を利用して計算を行っているため、3 次元環境においてもノード間距離を求めて利用することで 2 次元集約型 SOL と同様の方式を利用出来る。本章ではその中で 3 次元にするにあたって変更された部分について記述する。

### 4.1 共通 1 次近傍群領域判定によるトポロジ矛盾検知

位置推定精度の劣化の主要因である推定トポロジの折れ曲がり、各ノードからの 1 次近傍と 2 次近傍の距離によるトポロジ矛盾判定 (図 3) で検知できない場合がある。

図 5(a) に検知不可である折れ曲がりの場合を示す。この共通 1 次近傍群領域判定によるトポロジ矛盾検知は 3 次元で行っているため実際には通信範囲は球体、領域を分け

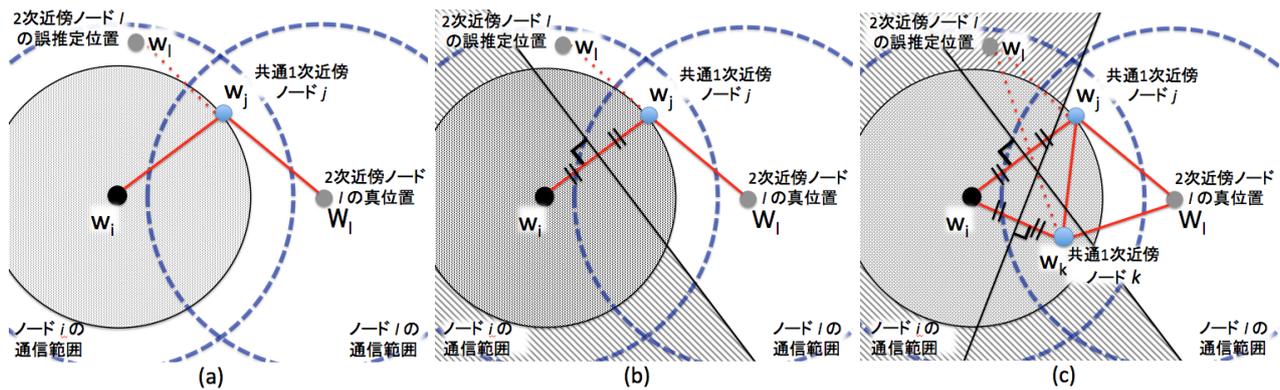


図 5 推定ノードの矛盾領域

る境界は面となるが、紙面の都合上、図 5 は 2 次元に次数を落として説明する。

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、 $w_l$  は折れ曲がりトポロジの要因となる。この場合、 $w_i$  と  $w_j$  を基準点とする  $w_l$  のトポロジ矛盾判定は  $w_l$  のトポロジ矛盾を検知できる範囲  $|w_l - w_i| \leq |w_j - w_i|$  の外にある（トポロジ矛盾なしの条件を満たす）ため、 $w_l$  による折れ曲がりトポロジを十分に検知可能とする領域をカバーできていない。従って、折れ曲がり推定トポロジの検知可能性を高めるためトポロジ矛盾判定の適応領域の拡大を図る。

図 5(b) に示すように、基準点  $w_i$  と  $w_j$  において、線分  $\overline{w_j - w_i}$  の中心において  $\overline{w_j - w_i}$  を法線とする平面を用いて  $w_i$  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 等分面の左側が  $w_i$  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、折れ曲がりトポロジの検知領域を拡大するため、図 5(c) に示すように、ノード  $j$  に加えて、ノード  $i$  とノード  $l$  の共通の 1 次近傍ノード  $k$  基準点として用いて、その推定位置  $w_k$  と  $w_i$  によりそれぞれに近い領域に 2 分割し、 $w_i$  に近い領域にある ( $|w_l - w_i| \leq |w_k - w_i|$ ) 場合トポロジ矛盾と判定する。以上のように、全ての共通 1 次近傍ノードを順次選択していき判定を繰り返すことでトポロジ矛盾の検知範囲を拡大し、折れ曲がり検知の可能性を高める。

この判定の実施結果において式 (9) を満たさない場合、トポロジ矛盾と判定する。  $A$  は共通 1 次近傍群領域判定を行った回数、 $a$  はトポロジ矛盾の発生回数、 $\beta$  は判定閾値である。

$$\frac{a}{A} \leq \beta. \quad (9)$$

## 4.2 絶対座標変換

SOL の推定位置はネットワーク内における相対位置である。この相対位置をアンカーノードの真位置と推定位置を用いて絶対位置へ変換する。アンカーノードの真位置  $W_A = (X_A, Y_A, Z_A)$  は推定位置  $w_A = (x_A, y_A, z_A)$  を用いて以下のように表される。

$$\begin{aligned} X_A &= ax_A + by_A + cz_A + t_x \\ Y_A &= dx_A + ey_A + fz_A + t_y \\ Z_A &= gx_A + hy_A + iz_A + t_z \end{aligned} \quad (10)$$

4 つのアンカーノードから構成される連立方程式 (10) から 12 個の係数  $a, b, c, t_x, d, e, f, t_y, g, h, i, t_z$  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  $w_i = (x_i, y_i, z_i)$  から絶対座標  $\hat{w}_i = (\hat{x}_i, \hat{y}_i, \hat{z}_i)$  へ変換される。

$$\begin{pmatrix} \hat{x}_i \\ \hat{y}_i \\ \hat{z}_i \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c & t_x \\ d & e & f & t_y \\ g & h & i & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \quad (11)$$

## 5. 評価方式

### 5.1 相対位置評価と絶対位置評価

以下の 2 方式での比較評価を行う。

- 2 次元集約型 SOL
- 3 次元集約型 SOL

評価は、相対位置評価と絶対位置評価の 2 通り行う。相対位置評価は推定されたノード位置により構成されるネットワークの形状（推定ネットワーク形状）とオリジナルネットワークの形状の相似性を、次の式により評価する。

$$r_{ij} = \frac{d_{ij}}{D_{ij}}, \quad \bar{r} = \frac{1}{|N|C_2} \sum_{i=1}^{|N|-1} \sum_{j=i+1}^{|N|} r_{ij} \quad (12)$$

$$V[r_{ij}] = \frac{1}{|N|C_2} \sum_{i=1}^{|N|-1} \sum_{j=i+1}^{|N|} (1 - r_{ij}/\bar{r})^2 \quad (13)$$

$d_{ij}$  は推定位置におけるノード  $i$  とノード  $j$  の距離（推定距離）、 $D_{ij}$  はオリジナルのネットワークにおけるノード

$i$  とノード  $j$  の距離,  $N$  は位置推定ノードの集合,  $|N|$  はノード数 (集合  $N$  の要素数) を示す.  $V[r_{ij}]$  が 0 に近づけば推定ネットワーク形状はオリジナルのネットワーク形状と相似となる. すなわち,  $V[r_{ij}]$  が 0 の場合, 推定ネットワーク形状はオリジナルのネットワーク形状に一致する.

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総和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 (14) のように求める.

$$Err_{ave} = \frac{1}{|N|} \sum_{i=1}^N |W_i - w_i| \quad (14)$$

$W_i$  はノード  $i$  の真位置,  $w_i$  は推定位置を示す.

## 5.2 評価結果

表 1 シミュレーション条件

評価条件	2D	3D
フィールド範囲	1.0 × 1.0	1.0 × 1.0 × 1.0
通信半径	0.23	0.23
アンカノード数	3	4
位置修正回数	300	300
評価トポロジ数	15	15

シミュレーションの評価条件を表 1 に示す. この条件でシミュレーションを行った結果のジオメトリ相似性が図 6, 位置推定誤差が図 7, である. また集約型 SOL において重要な要素は近傍ノード数であるため, 比較基準として評価結果の横軸に近傍ノード数を設定することで, 2次元と3次元の次数の違いを正規化している.

まず図 6 のジオメトリ相似性について見てみると, 同程度の近傍ノード数において 3次元集約型 SOL は, 他方式と比べて高精度であった 2次元集約型 SOL と比較してほぼ同等の精度となる. また, 近傍ノード数が増える毎に 2次元集約型 SOL と同様に  $V[r_{ij}]$  は極めて 0 に近づいている. 従って, 3次元集約型 SOL の位置推定はネットワークのオリジナルジオメトリを高い精度で再現出来ていることが分かる. また 3次元にすることで 2次元と同じノード数では空間におけるノードの密度が下がることから近傍ノード数は減少するが, 図 6 の近傍ノード数増加毎の精度上昇を見るとどちらも同じ傾向が見られ, およそ近傍ノード数 30 前後の時点で 2次元集約型 SOL および 3次元集約型 SOL 共に,  $V[r_{ij}]$  は 0.01 程度まで収束していることが分かる.

次に図 7 の位置推定誤差を見てみると, 先ほどのジオメトリ相似性と同様に, 3次元集約型 SOL の近傍ノード数毎の位置推定誤差や近傍ノード数増加による精度の上昇は 2次元集約型 SOL と比較しても大きな差が見られず, 高い精度での位置推定が行えていることが分かる. また位置推定誤差についても 2次元集約型 SOL と同様の傾向が見られ, 近傍ノード数が 30 前後で  $Err_{ave}$  は 0.05 前後, 近傍ノード数が 50 前後で  $Err_{ave}$  は 0.04~0.03 程度となる.

以上のことから, 近傍ノード数が同程度であれば, 3次元集約型 SOL は 2次元集約型 SOL と比較してもほとんど精度の劣化は無いと言える.

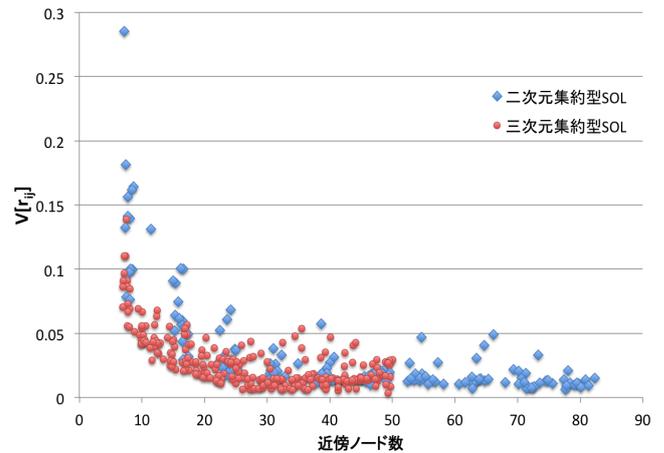


図 6 近傍ノード数毎のジオメトリ相似性

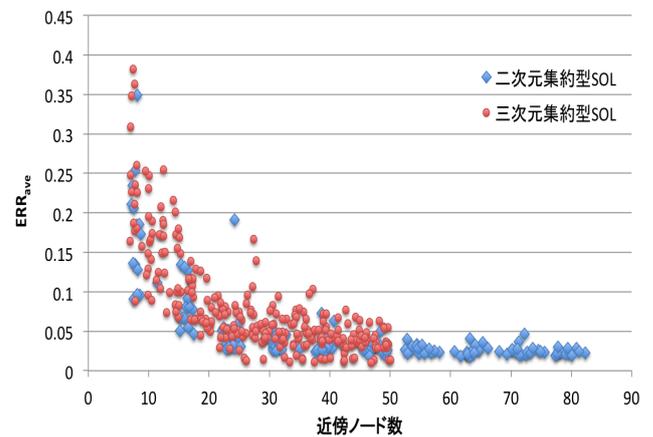


図 7 近傍ノード数毎の位置推定誤差

次にこの 3次元集約型 SOL のシミュレーションで位置推定を行ったネットワークのオリジナルジオメトリと推定ジオメトリとを比較した図を見てみる. ノード数 100 の場合のオリジナルジオメトリと推定ネットワークジオメトリをそれぞれ図 8 と図 9 に, ノード数 200 の場合のオリジナルジオメトリと推定ネットワークジオメトリをそれぞれ図 10 と図 11 に示す. ノード数 100 個の場合, 各ノードの近傍ノード数はおよそ 10 個程度となるため, シミュレーション結果で示したグラフを見てみると, 位置推定精度としては 0.2 程度の誤差があり, 大きな値である. しかし, 図 8 と図 9 を比較すると分かるように, ネットワークのジオメトリの特徴は良く捉えられており, 位置推定精度の誤差が高くと, その有用性が高いと考える. 次にノード数 200 個の場合, 各ノードの近傍ノード数は 15 個程度となり, ノード数 100 個の場合よりも更に推定ジオメトリがオリジナルジオメトリを再現出来ている事が分かる. 以上の 2つ

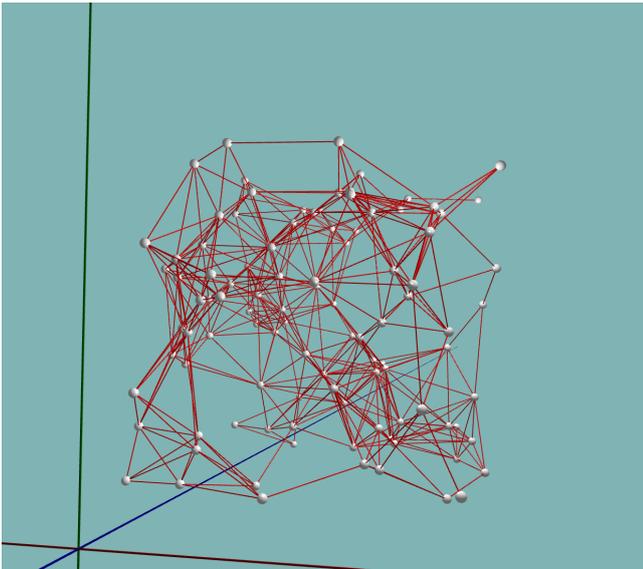


図 8 100 ノードでのネットワークのオリジナルジオメトリ

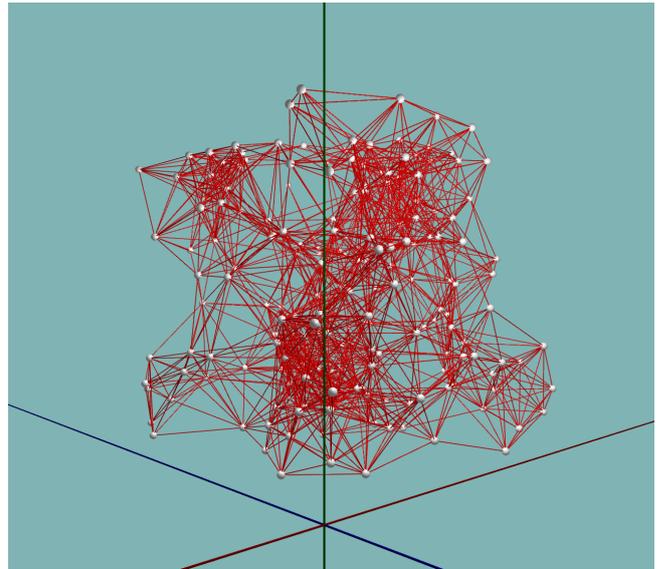


図 10 200 ノードでのネットワークのオリジナルジオメトリ

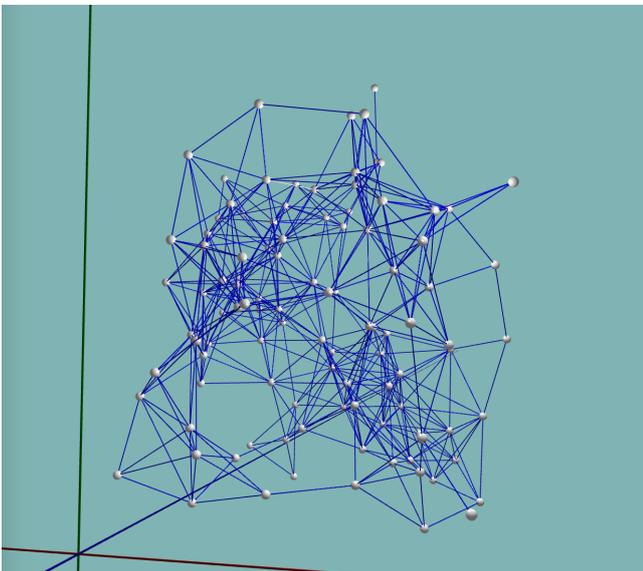


図 9 100 ノードでの推定ネットワークジオメトリ

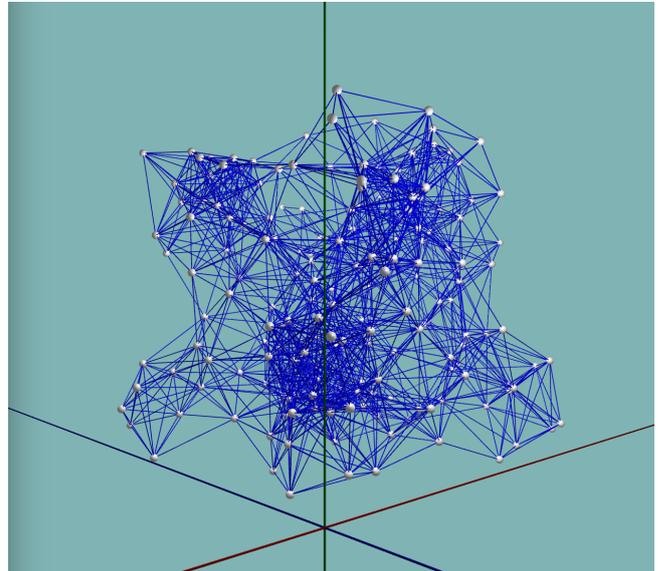


図 11 200 ノードでの推定ネットワークジオメトリ

の図を見てからも分かるように、推定されたネットワークジオメトリはオリジナルのネットワークジオメトリの特徴を再現できており、3次元集約型 SOL は 2次元集約型 SOL と同様に高い精度でネットワークのジオメトリを再現できていることが確認出来た。

以上のことから、集約型 SOL の位置推定は 3次元環境においても有効であると考えられる。

## 6. まとめ

本論文は、クラウド環境を想定した集約型 SOL の 3次元環境への適用を提案した。また、3次元集約型 SOL のシミュレーション評価から、3次元環境においても 2次元環境と同程度の高い位置推定精度を維持可能であることを確認した。

今後は 3次元集約型 SOL の障害物混在環境での精度評

価および実環境での精度評価を検討する予定である。

## 参考文献

- [1] Kortuem, G., Kawsar, F., Fitton, D. and Sundramoorthy, V., "Smart objects as building blocks for the Internet of things," *IEEE Internet Computing*, Vol.14, No.1, pp.44-51 (2010).
- [2] Geng, Wu., Talwar, S., Johnsson, K., Himayat, N. and Johnson, KD., "M2M : From mobile to embedded internet," *IEEE Communications Magazine*, Vol.49, No.4, pp.36-43 (2011).
- [3] 大野翔平, 安達直世, 滝沢泰久, "無線センサネットワークにおける自己組織化位置推定方式の提案," *情報処理学会論文誌*, Vol.53, No.7, pp.1774-1782, (2012).
- [4] Kitanouma, T.; Takashima, Y.; Adachi, N.; Takizawa, Y., "Cloud-based Self-Organizing Localization for wireless sensor networks in mixture environments of LOS and NLOS," *Proc. IEEE IWCMC 2015*, pp.1230-1235 (2015).
- [5] Harter, A., Hopper, A., Steggles, P., Ward, A., and

- Webstar, 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., Odziemzyk, W., and Nagorski, K., “Investigation of Practical and Theoretical Accuracy of Wireless Indoor Positionings 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] Niculescu, D. and Nath, B. “DV-based positioning in ad hoc networks,” *Telecommun.Syst*, Vol.22, No.1-4, pp267-280, (2003).
- [12] Zhou, Z., Xiao, M., Liu, L., Chen,Y. and Lv,J., “An Improved DV-HOP Localization Algorithm,” *Proc. ISISE 2009*, pp.598-602, (2009).