

# Region of Interest Growing Neural Gasに関する検討

## Study of Region of Interest Growing Neural Gas

○戸田 雄一郎<sup>1</sup>, 松野 隆幸<sup>1</sup>, 見浪 譲<sup>1</sup>  
 ○Yuichiro Toda<sup>1</sup>, Takayuki Matsuno<sup>1</sup>, Mamoru Minami<sup>1</sup>

<sup>1</sup>岡山大学大学院  
<sup>1</sup>Okayama University

**Abstract:** This paper focuses on a real-time topological structure learning method for a 2D/3D point cloud. First of all, we explain a Region of Interest Growing Neural Gas (ROI-GNG) proposed in our previous research ROI-GNG realize concentrated/distributed sensing in real-time. Next, we show an experimental result for discussing the effectiveness of ROI-GNG.

### 1. はじめに

自律ロボットにとって、実時間で環境を認識・知覚することは、重要な技術の一つである。環境認識技術の中でも、近年では、RGB-D カメラや Lidar などの発展に伴い、環境の 3 次元空間認識技術に関する開発・研究が多く行われており、自律ロボットの安全な移動や作業を実現するためにも、様々な分野で期待されている [1-6]。我々は、これまで、3 次元点群処理における位相構造学習手法として、ある属性値を持つ対象物に対して、精度の高い環境認識を行うために、詳細なモデル化を行うと同時に、それ以外の属性を持つデータに対しては、周囲を探索することが可能なように、疎なモデル化を行うといった、局所集中的探索と大域分散的探索を同時に実現する手法として、自己組織化手法の 1 つであり、空間的な位相構造の学習が可能な Growing Neural Gas (GNG) [1] を用いた Region of Interest GNG (ROI-GNG)を提案してきた [2]が、点群データに対しての十分な検証は、これまで行われてこなかった。本稿では、ROI-GNG の検証を行うために、いくつかのシミュレーションデータに対して ROI-GNG を適用することで、その有効性の検証を行っていく。

### 2. Region of Interest Growing Neural Gas

これまで、我々は、ノードの削除速度を制御可能であり、関連重要度と呼ばれる複数の特徴量を学習しながら空間的な位相構造を保存可能な手法として、改良型 GNG-U (GNG-U2) を提案した。GNG-U2 では、変化の激しい時系列データに対して適用可能な手法となっており、RGB-D カメラなどの 3 次元点群データに対して適用し、その追従性の向上や実時間における領域分割手法などを実現してきた。さらに、GNG-U2 を発展させた学習手法として、集中的探索と分散的探索の概念を GNG-U2 に取り入れた手法として、ROI-GNG を提案し、ある属性値を持つ対象物に対して、詳細な位相構造を学習、それと同時に、それ以外の属性を持つデータに対して、疎な位相構造の学習を行うことが可能なアルゴリズムを提案してきた。ここではまず、基本的な ROI-GNG の学習アルゴリズムを以下に示す。

$h_i$ : The  $n$ th dimensional vector of a node

$A$ : A set of nodes

$N_i$ : A set of nodes connected to the  $i$ th node

$c$ : A set of edges

$g_{ij}$ : Age of the edge between the  $i$ th and  $j$ th nodes

Step0. 初期化として、二つのノードの参照ベクトル  $w_1$  と  $w_2$  をランダムに生成し、結合関係  $c_{1,2}=1$ 、エッジの年齢  $g_{1,2}=0$  とする。

Step1. 入力データ  $v$  をランダムに取得する。

Step2. 入力データ  $v$  に対する勝者ノード  $s_1$  と第 2 勝者ノード  $s_2$  を選択する。

$$\begin{aligned} s_1 &= \arg \min_{i \in A} \|w * (v - h_i)\| \\ s_2 &= \arg \min_{i \in A \setminus s_1} \|w * (v - h_i)\| \end{aligned} \quad (1)$$

ここで、\*は、要素ごとの積を表す。

Step3. ノード  $s_1$  について入力データ  $v$  との二乗誤差を積算誤差  $E_{s_1}$  に加算する。また、ノード  $s_2$  と  $s_1$  の二乗誤差の差を Utility 値  $U_{s_1}$  に加算する

$$\begin{aligned} E_{s_1} &\leftarrow E_{s_1} + \|w * (v - h_{s_1})\|^2 \\ U_{s_1} &\leftarrow U_{s_1} + \|w * (v - h_{s_2})\|^2 - \|w * (v - h_{s_1})\|^2 \end{aligned} \quad (2)$$

Step4. ノード  $s_1$  及びノード  $s_2$  と結合関係があるノードの参照ベクトルを更新する。ただし、 $\eta_1$  と  $\eta_2$  を学習係数とする ( $\eta_1 > \eta_2$ )。

$$\begin{aligned} h_{s_1} &\leftarrow h_{s_1} + \eta_1 \cdot (v - h_{s_1}) \\ h_j &\leftarrow h_j + \eta_2 \cdot (v - h_j) \quad \text{if } c_{i,j} = 1 \end{aligned} \quad (3)$$

Step5. エッジの年齢を 0 にリセットする ( $g_{s_1, s_2} = 0$ )。また、ノード  $s_1$  と  $s_2$  との間にエッジが存在しなければ、新たにエッジを作成する ( $c_{s_1, s_2} = 1$ )。

Step6. ノード  $s_1$  と結合関係のある全てのエッジの年齢をインクリメントする。

$$g_{s_1, j} \leftarrow g_{s_1, j} + 1 \quad \text{if } g_{s_1, j} = 1 \quad (4)$$

Step7. 事前に設定した閾値  $g_{\max}$  を超える年齢のエッジを削除する ( $c_{s_1, s_2} = 0$ )。その結果、他のノードと結合関係を持たないノードが現れた場合は、当該ノードを削除する。

Step8. データ入力が  $\lambda$  回ごとに、次の操作を行う。

i. 積算誤差が最大のノード  $u$  を選択する。また、最小の Utility 値を持つノード  $l$  を選択する

$$u = \arg \max_{i \in A} E_i$$

$$l = \arg \min_{i \in A} U_i$$
(5)

ii. ノード  $u$  と結合関係のあるノードの中で最も積算誤差の大きいノードを  $f$  とし,  $u$  と  $f$  を結合するエッジを 2 分するようにノード  $r$  を挿入する.

$$h_r = 0.5 \cdot (h_u + h_f) \quad (6)$$

iii. 最大積算誤差と最小の Utility 値が以下の式を満たす場合, ノード 1 の削除を行なう.

$$E_u / U_l > k \quad (7)$$

iv. 次に, ノード  $u, f$  間のエッジを削除し ( $c_{uf}=0$ ), ノード  $u, r$  及び  $r, f$  間にエッジを追加する ( $c_{ur}=1, c_{rf}=1$ ).

v. ノード  $q, f$  の積算誤差を以下の式により更新する.

$$E_u \leftarrow E_u - \alpha E_u$$

$$E_f \leftarrow E_f - \alpha E_f \quad (8)$$

vi. ノード  $u, f$  の積算誤差の平均をノード  $r$  の積算誤差とする.

$$E_r = 0.5 \cdot (E_u + E_f) \quad (9)$$

Step9. GNG へのデータ入力が  $\kappa (< \lambda)$  回ごとに, Step 8 における, i と iii を行うことによって, ノード数の調整を行なう.

Step10. 全てのノードの誤差と Utility 値を減らす.

$$E_i \leftarrow E_i - \beta_i^t E_i \quad (\forall i \in A)$$

$$U_i \leftarrow U_i - \chi_i^t U_i \quad (\forall i \in A) \quad (10)$$

ここで,  $\beta_i^t, \chi_i^t$  は, それぞれ時刻  $t$  における  $i$  番目のノードの積算誤差と Utility 値の減衰率を表す.

Step11. 全てのノードの減衰率を更新する.

$$\beta_i^t = \begin{cases} b_{low} & \text{if the } i\text{th node has objective feature} \\ b_{high} & \text{otherwise} \end{cases} \quad (11)$$

$$\chi_i^t \leftarrow r^t \cdot \beta_i^t \quad (12)$$

ここで,  $r^t$  は, 時刻  $t$  における  $i$  番目のノードの積算誤差  $\beta_i^t$  の減衰率に対する Utility 値の減衰率の大きさを表す.

Step12. 終了条件が満たされない場合は, step2 に戻る

このように, ROI-GNG では, GNG-U2 のアルゴリズムに基づき積算誤差と Utility 値の減衰率を可変とし, Step11において, 減衰率を調整することで, 実時間で集中・分散的探索が可能な位相構造の学習が可能となっている.

### 3. 実験結果

本章では, ROI-GNG を 2 次元の点群データに適用することによって, 検証を行なっていく. [2]においては, 図 1(a)に示すように各色が異なるようにリングを配置し, クラスタができるやすいデータにおいて検証を行なったが, 本稿では(b)に示すように, 各色のデータ (青: 5000 個, 緑: 5000 個) が入り交じったもので ROI-GNG の検証を行なっていく. また, 各パラメータは, [2]で用いたものと同様のものを使用し, 1000 回の学習を行う. 0-1000 回までは緑色に対して密な学習を行い, 1001- 2000 回までは青に対して密な学習を行うように設定した. 図 2 に実験結果を示す. 図 2 の

各色の線は, それぞれの色のノード数を表しており, 実験の設定通り, 対象となる属性データに対して密な学習ができていることが分かる. しかしながら, 図 1(a)を用いた[2]結果と比較すると, ノード数に大きな振動が見られている. これは, 色の境目の部分において, 減衰率が振動し, 学習が安定できなかつたためだと考えられる.

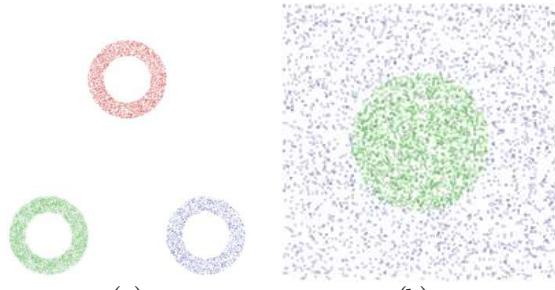


図 1 シミュレーションデータ

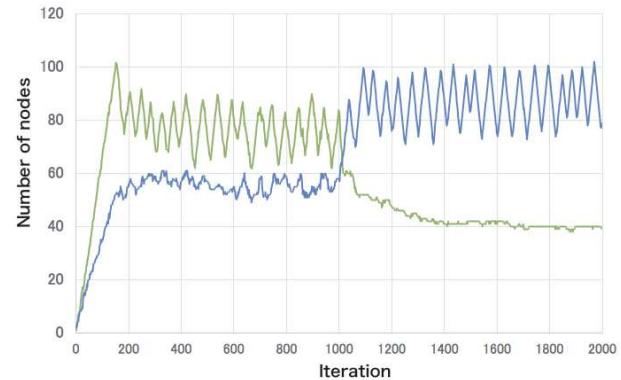


図 2 実験結果

### 4. おわりに

本稿では, これまで我々が提案してきた ROI-GNG をクラスタリングしづらいデータに適応することで, ROI-GNG の有効性の検証を行なった. 今後は, 実験において生じたノード数の振動を押さえるために, 減衰率の更新方法の再考を行なっていく予定である.

### 5. 謝辞

本研究は JSPS 科研費 JP19H04190 の助成を受けたものです.

### 参考文献

- [1] B. Fritzke: A growing neural gas network learns topologies; Advances in Neural Information Processing Systems, Vol. 7, pp. 625-632, 1995.
- [2] 戸田, "GROWING NEURAL GAS を用いた集中/分散的センシングに基づく 3 次元空間の位相構造学習", 第 24 回ロボティクスシンポジア, pp.90-93, 2019

### 連絡先

戸田 雄一郎

E-mail: ytoda@okayama-u.ac.jp