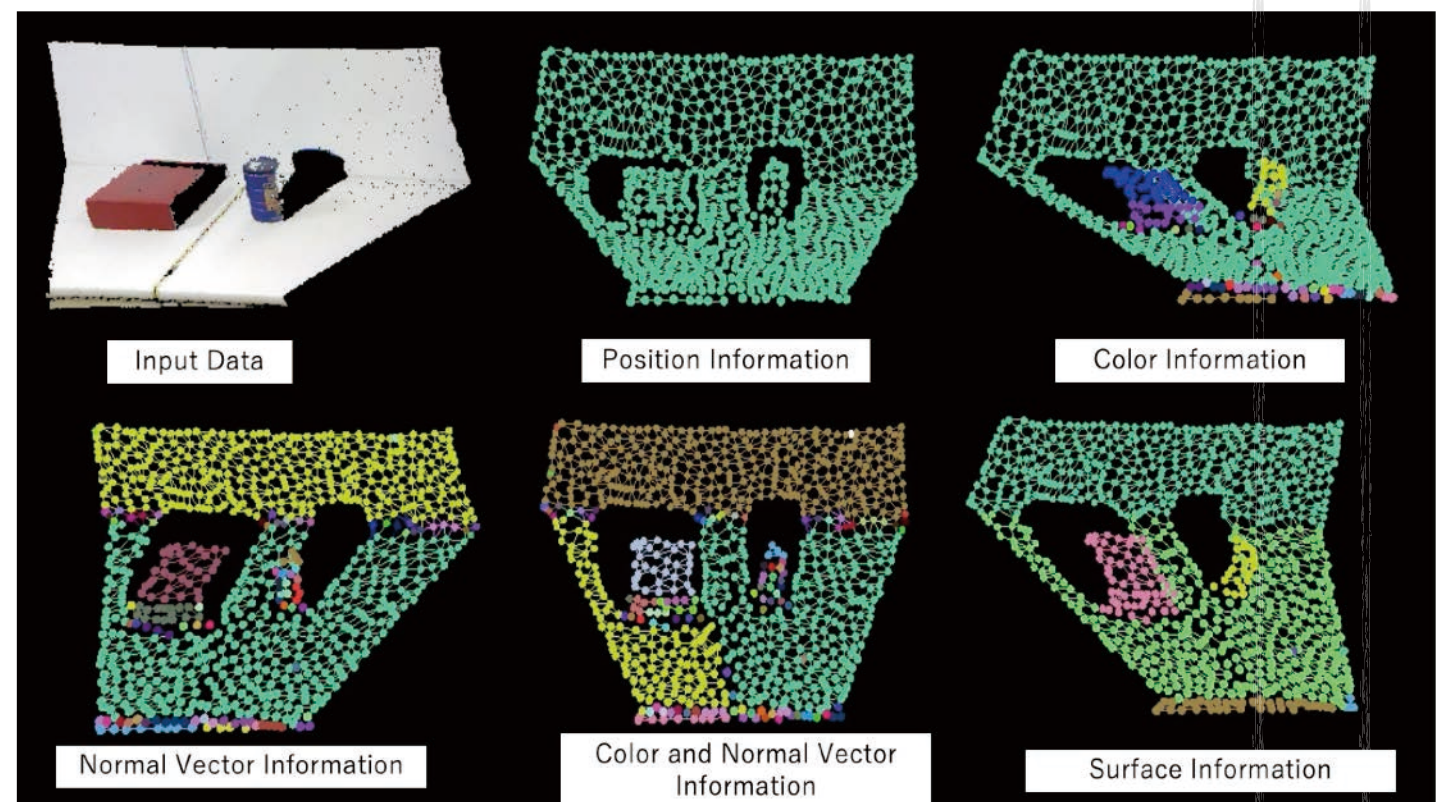
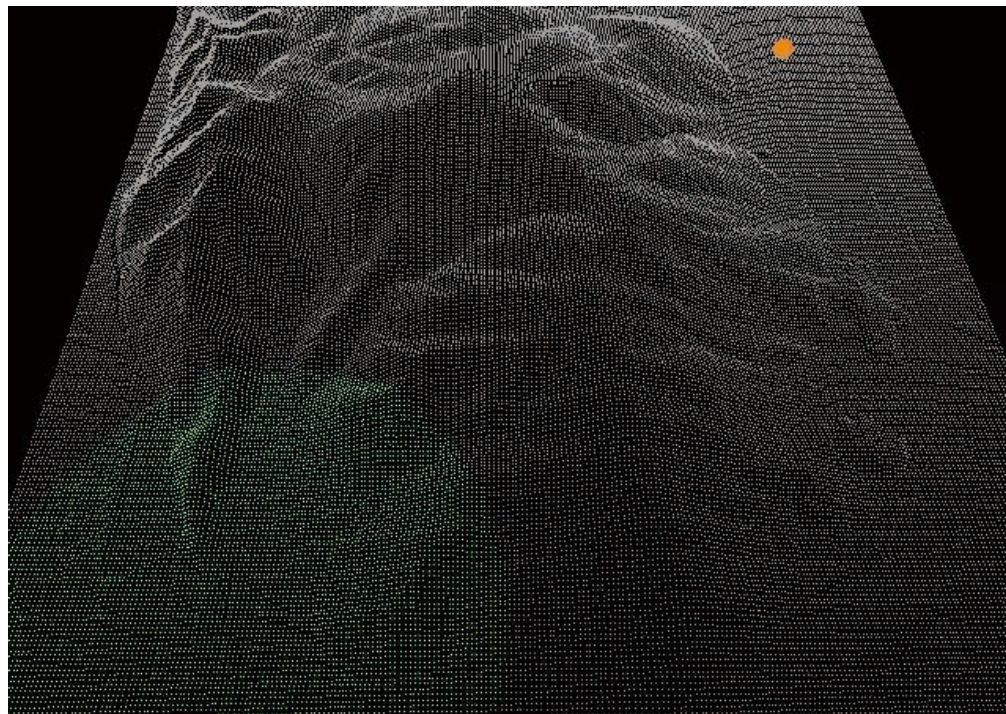


Growing Neural Gasの基礎と点群処理

岡山大学 学術研究院自然科学学域

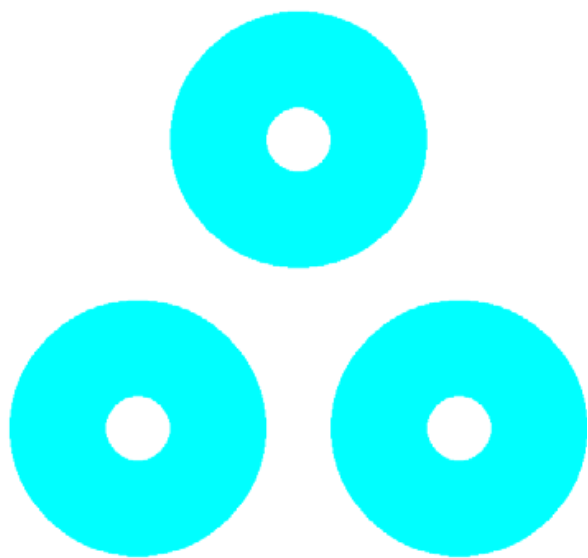
助教 戸田 雄一郎



Growing Neural Gas

- ・自己組織化マップの拡張手法であるGrowing Neural Gasは、エッジの追加や削除を行う手法が組み込まれており、冗長なエッジやノードが発生しにくい手法
- ・そのため、与えられたデータ構造に従った位相構造の学習及び位相構造に基づくクラスタリングが可能となる

	Node	Edge Addition	Edge Deletion	Triangulation
Self-Organizing Maps	const	×	×	×
Neural Gas	const	-		×
NG with CHL	const	○	×	×
Growing Cell Structure	growth	○	×	○
Growing Neural Gas	growth	○	○	×



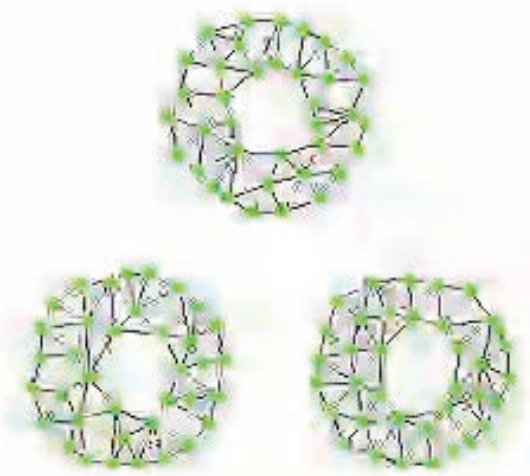
Input data



SOM



GCS

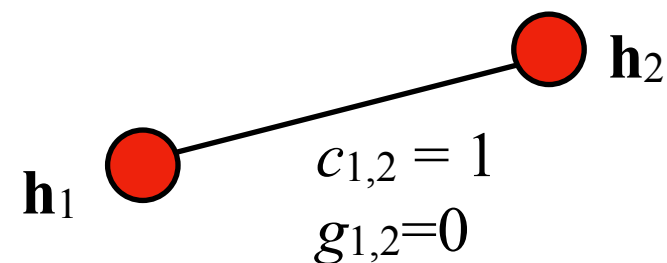


GNG

Growing Neural Gas

初期化

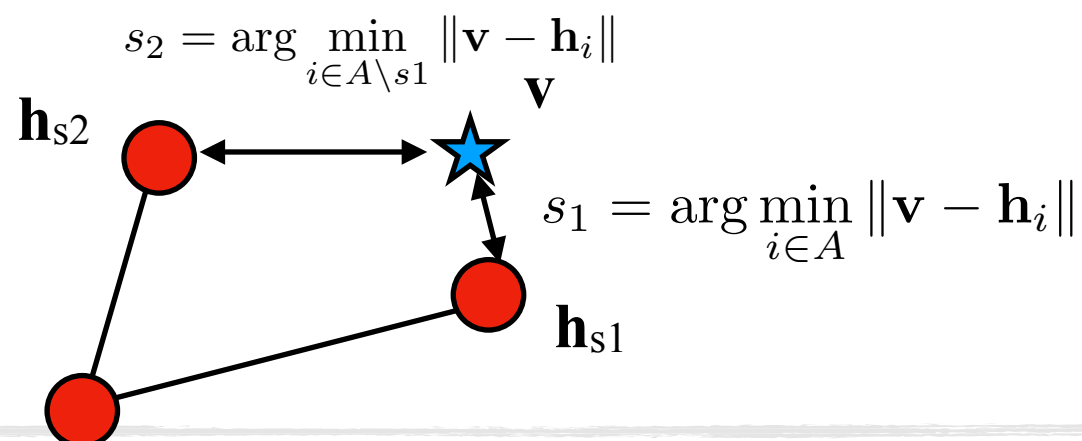
二つのノードの参照ベクトル $\mathbf{h}_1$ と $\mathbf{h}_2$ をランダムに生成し, 結合関係 $c_{1,2} = 1$, エッジの年齢 $g_{1,2}=0$ とする.



勝者ノードの選択

入力ベクトル $\mathbf{v}$ を計測データからランダムに1つ取得する.

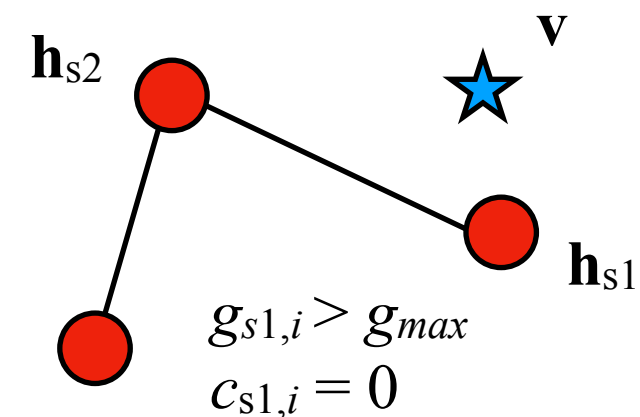
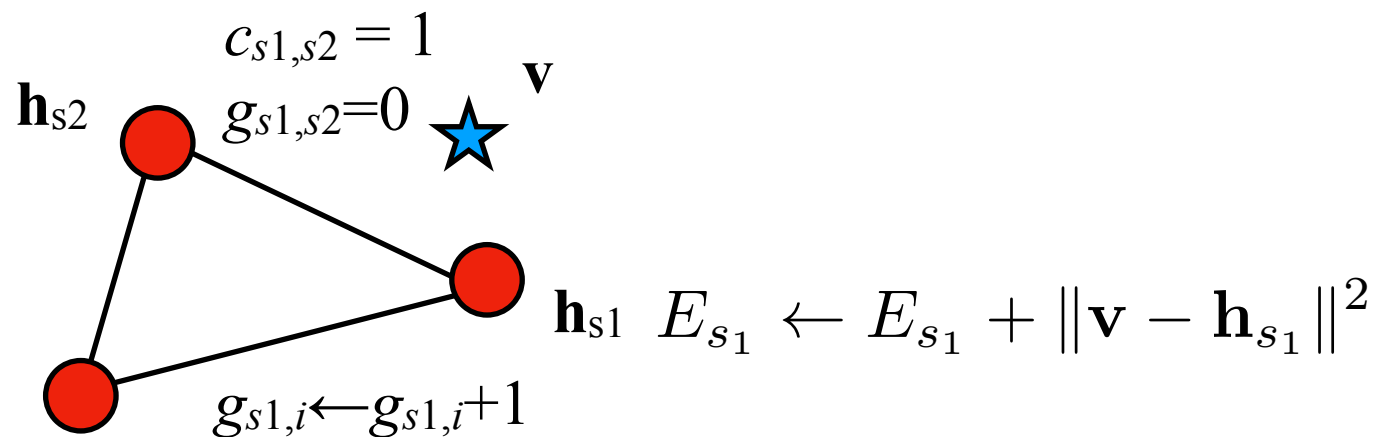
入力ベクトル $\mathbf{v}$ に対する第1勝者ノード s_1 と第2勝者ノード s_2 を選択する.



Growing Neural Gas

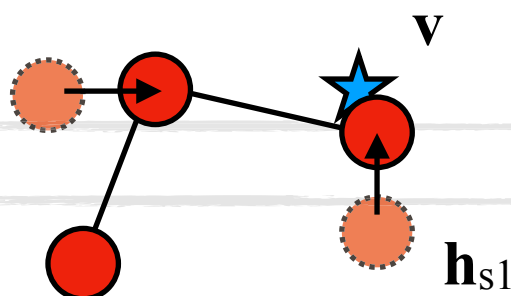
エッジと積算誤差の更新

- ・ ノード s_1 について入力データ $\mathbf{v}$ との二乗誤差を積算誤差 E_{s_1} に加算する.
- ・ ノード s_1 と結合関係がある全てのエッジの年齢をインクリメントする.
- ・ エッジの年齢を0にリセットし, 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.
- ・ 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 ($c_{i,j}=0$).
- ・ その結果, 他のノードと結合関係を持たないノードは, そのノードを削除する.



ノードの更新

ノード s_1 及びノード s_1 と結合関係があるノードの参照ベクトルを更新する.



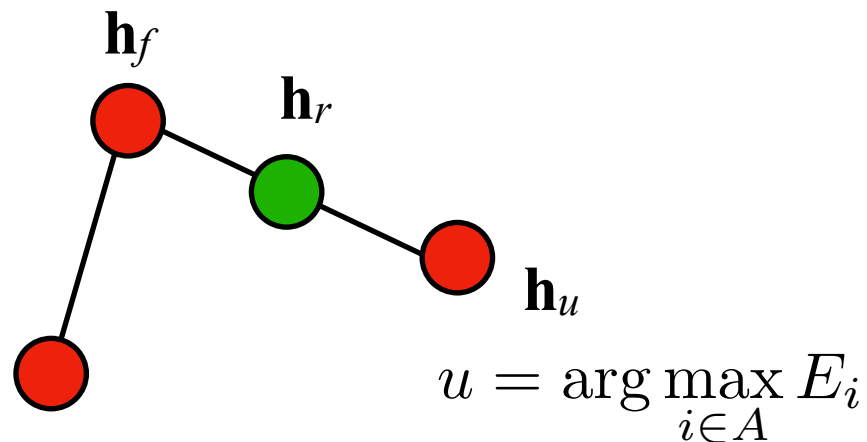
$$\mathbf{h}_{s_1} \leftarrow \mathbf{h}_{s_1} + \eta_1 (\mathbf{v} - \mathbf{h}_{s_1})$$

$$\mathbf{h}_j \leftarrow \mathbf{h}_j + \eta_2 (\mathbf{v} - \mathbf{h}_j) \quad \text{if } c_{s_1,j} = 1$$

Growing Neural Gas

ノードの追加(λ 回ごと)

- ・ 積算誤差が最大のノード u を選択する.
- ・ 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.
- ・ ノード u, f 間のエッジを削除し ($c_{u,f}=0$), ノード u, r 及び r, f 間にエッジを追加する ($c_{u,r}=1, c_{r,f}=1$).
- ・ ノード u, f の積算誤差を減衰率 α を用いて更新する.
- ・ 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.



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

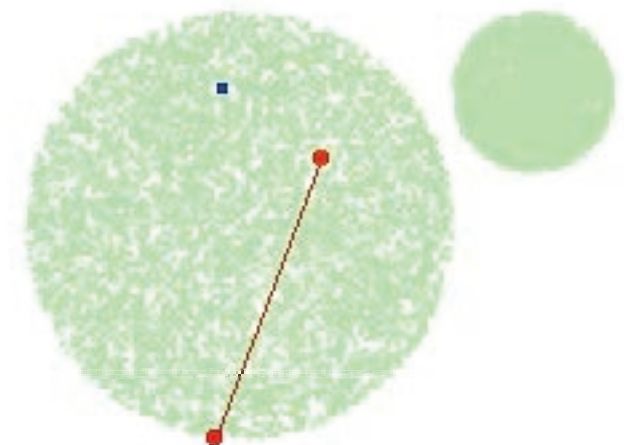
$$E_f \leftarrow E_f - \alpha E_f$$

$$E_r = 0.5(E_u + E_f)$$

積算誤差の減衰

全てのノードの誤差を減衰率 β により減衰させる.

$$E_i \leftarrow E_i - \beta E_i \quad (\forall i \in A)$$

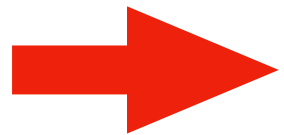


Growing Cell Structureにおけるノードの追加指標

- ・ 成長型の競合学習で、常にk-dimensionとなる位相構造を構築しながら成長していく学習アルゴリズム

GCSの目的

- ・ 位相構造と入力ベクトル空間に近い構造を持つように学習する
- ・ データの確率密度分布に近い構造を学習する



この2つの目的は同時に達成できるか？

GCSのノードの追加条件

- ・ GCSにおいては、積算されていくノードの追加指標を”resource”と呼び、以下の式において定義をしている

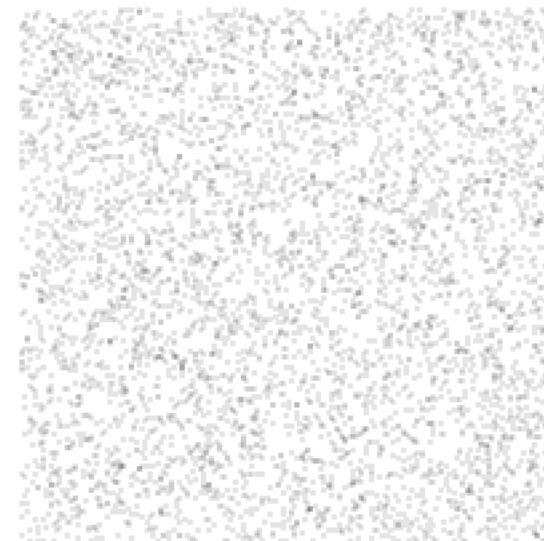
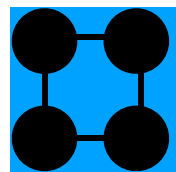
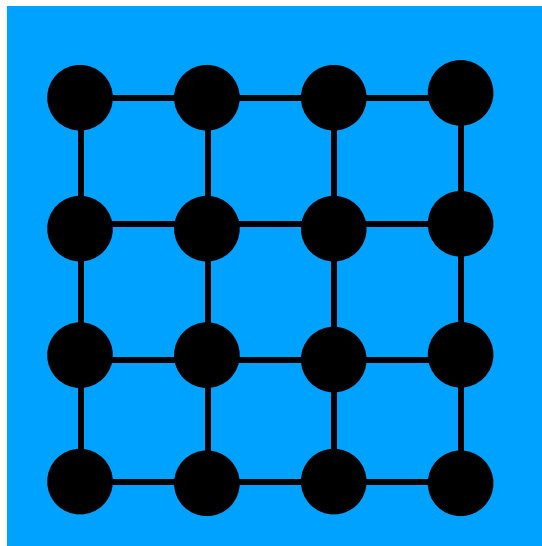
$$E_{s_1} \leftarrow E_{s_1} + \|\mathbf{v} - \mathbf{h}_{s_1}\|^2 \quad \text{or} \quad E_{s_1} \leftarrow E_{s_1} + 1$$

Growing Neural Gasと積算誤差

$$E_{s_1} \leftarrow E_{s_1} + \|\mathbf{v} - \mathbf{h}_{s_1}\|^2$$

GNGでは、入力ベクトルと第1勝者ノードの参照ベクトルの距離の二乗の累積値が、ノード追加の条件として使用される

積算誤差は妥当か？



データ数:5000

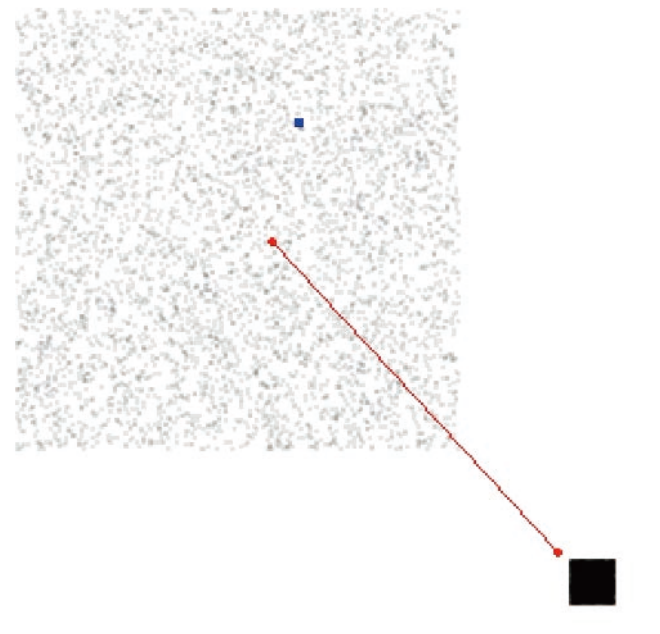
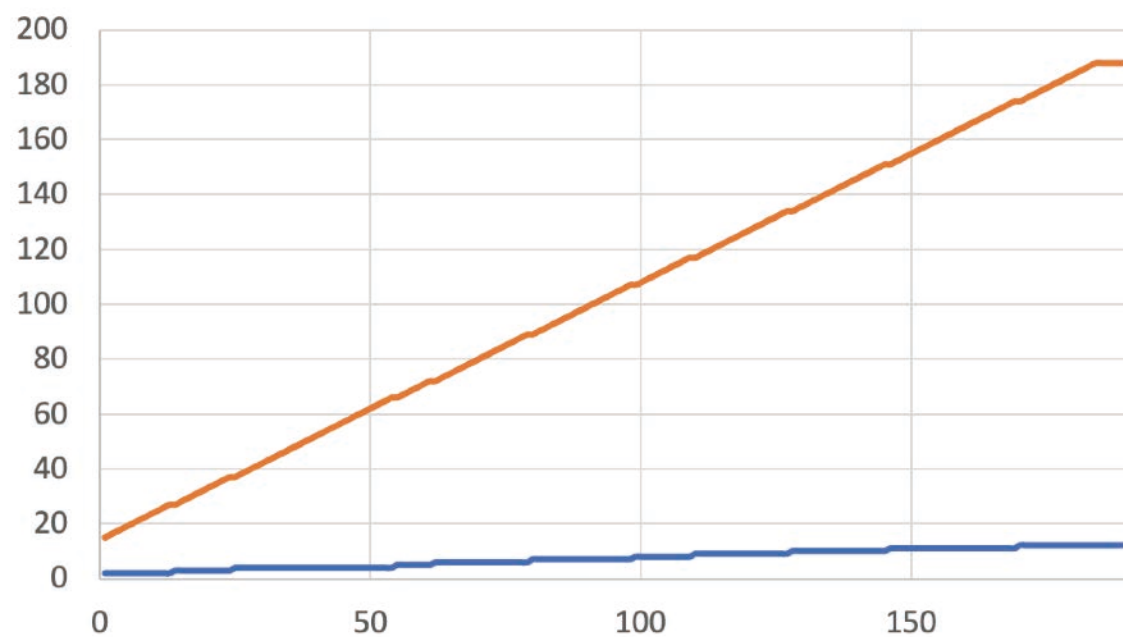


データ数:5000

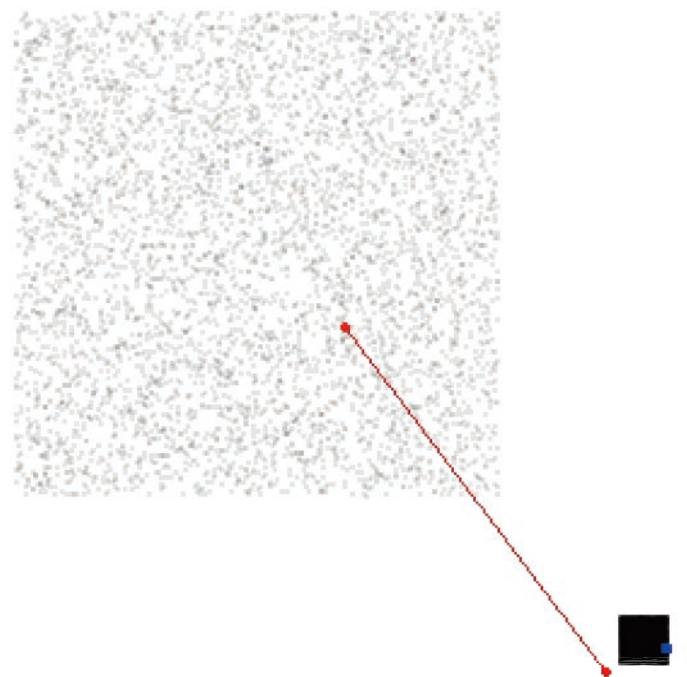
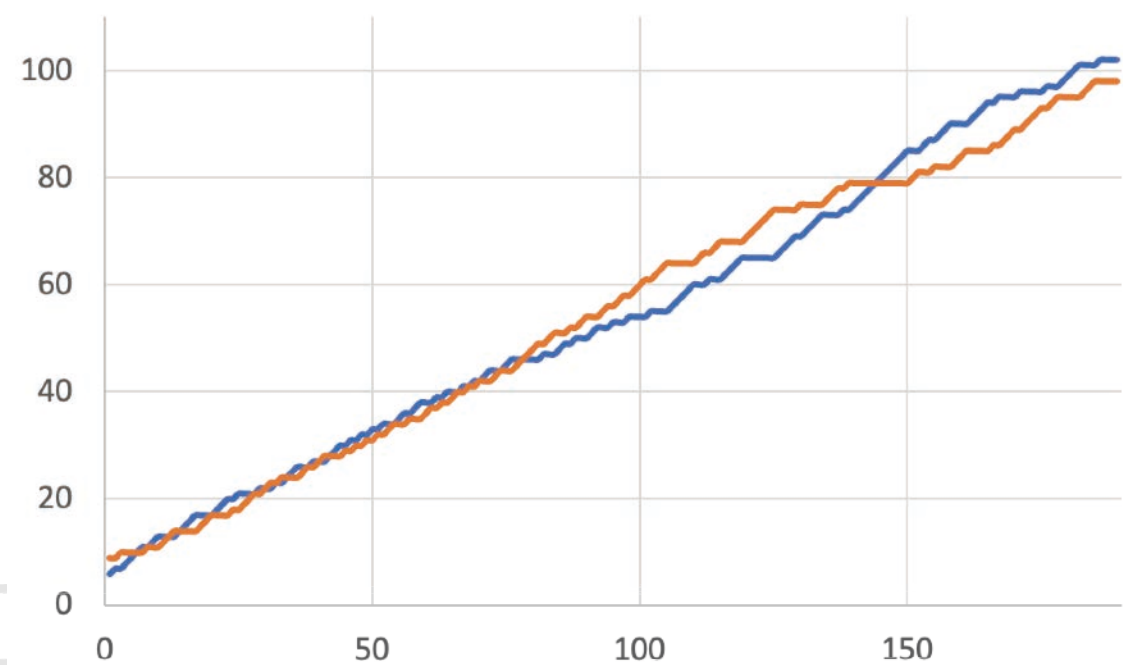
どのように、位相構造を学習したいかは目的によって異なる

Growing Neural Gasと積算誤差

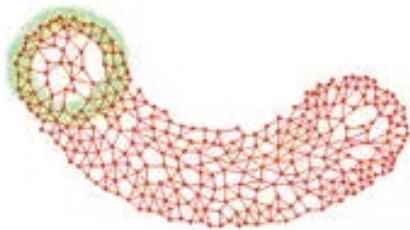
$$E_{s_1} \leftarrow E_{s_1} + \|\mathbf{v} - \mathbf{h}_{s_1}\|^2$$



$$E_{s_1} \leftarrow E_{s_1} + 1$$



Growing Neural Gasの問題点？



非定常データに対してGNGを適用すると現時刻にデータが存在しない位置にノードが残ってしまう

動的(非定常)データへの適用

問題点？

GNGのノードを記憶として捉える場合に悪くない性質であるが、時系列データを処理したい場合には困った性質となる

解決策

ノードを削除する基準を決めて、動的にノードの数をコントロールする学習アルゴリズムを考える

Growing Neural Gas with Utility

非定常データに適用するためにFritzkeによって提案されている
GNG with Utility (GNG-U) を用いる

削除のための指標

第1勝者ノードのユーティリティ値 U_{s1} を更新

$$U_{s1} \leftarrow U_{s1} + ||\mathbf{v} - \mathbf{h}_{s2}||^2 - ||\mathbf{v} - \mathbf{h}_{s1}||^2$$

ユーティリティ値の減衰

$$U_c \leftarrow U_c - \chi U_c \quad \rightarrow \quad \text{減衰率}\chi\text{により積算誤差と同様に減衰させる}$$

削除の基準

最小のユーティリティ値が低く、最大の積算誤差が大きい場合、ノードを削除する

$$q = \arg \max_c E_c$$

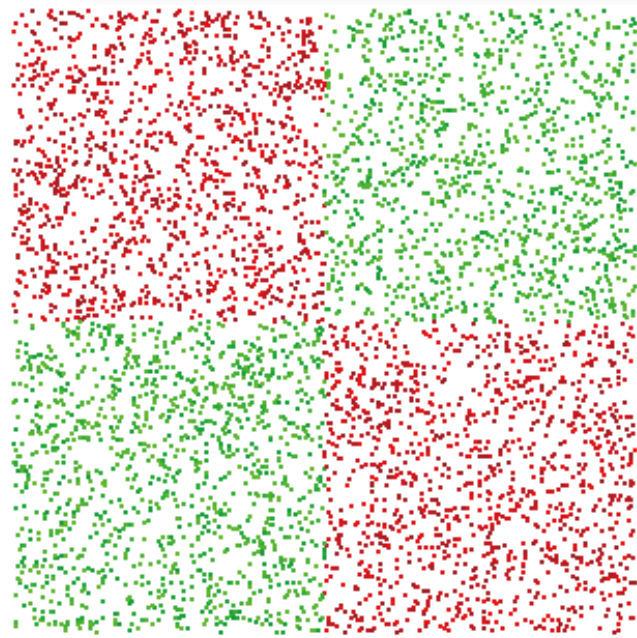
$$i = \arg \min_c U_c$$

$$E_q / U_i > k$$

基準は固定のしきい値 k によって決定される

Growing Neural Gasの問題点（その2）？

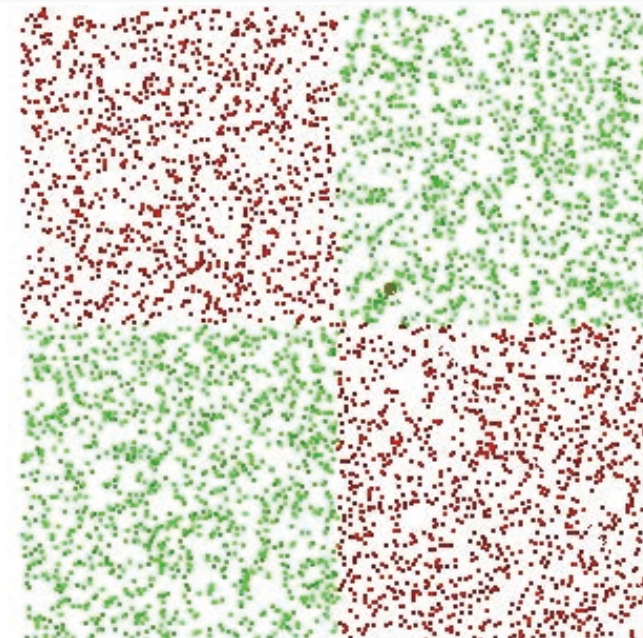
これまでの入力ベクトルは位置座標のみで構成されていた



色付き点群



どうなる？



適用結果

問題点？

入力ベクトルが異なる複数の属性で構成されている場合、位置情報の幾何的な空間構造を保持することができない

解決策①

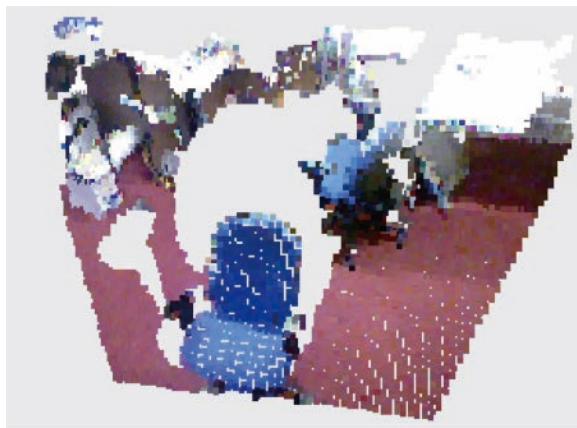
入力ベクトルの属性に重みを付けて、重要な要素を重点的に学習する方法論を考える

関連重要度を用いた色付き点群の学習

重み（関連重要度）のベクトルを定義し、勝者ノードの選択と積算誤差の計算時に重み付き距離尺度を用いるだけで良い

勝者ノードの選択

$$s_1 = \arg \min_{i \in A} \left\| \underline{w} * (v - h_i) \right\|$$
$$s_2 = \arg \min_{i \in A \setminus s_1} \left\| \underline{w} * (v - h_i) \right\|$$



Original 3D point cloud

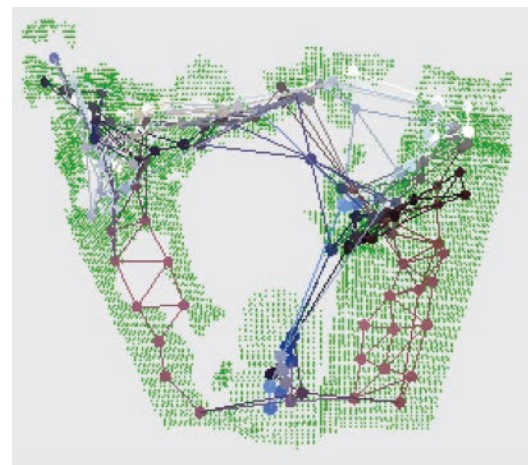
$$v_i = (v_{i,x}, v_{i,y}, v_{i,z}, v_{i,R}, v_{i,G}, v_{i,B})^T$$

積算誤差の計算

$$E_{s_1} \leftarrow E_{s_1} + \left\| \underline{w} * (v - h_{s_1}) \right\|^2$$

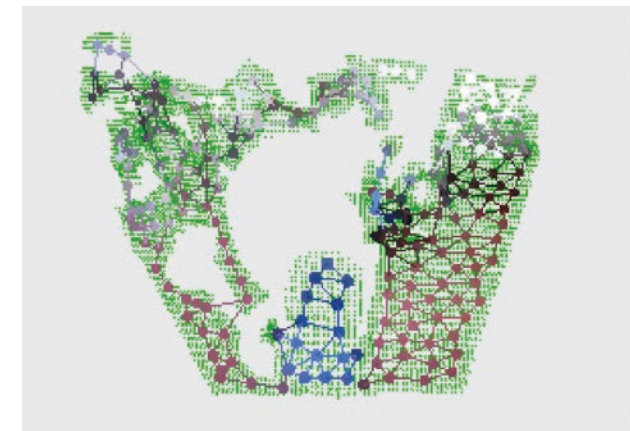
$$U_{s_1} \leftarrow U_{s_1} + \left\| \underline{w} * (v - h_{s_2}) \right\|^2 - \left\| \underline{w} * (v - h_{s_1}) \right\|^2$$

w : Related importance vector
 v : Input vector
 h_i : The i th node position
 $*$: Element wise production



Without Related importance

$$w = (1, 1, 1, 1, 1, 1)^T$$



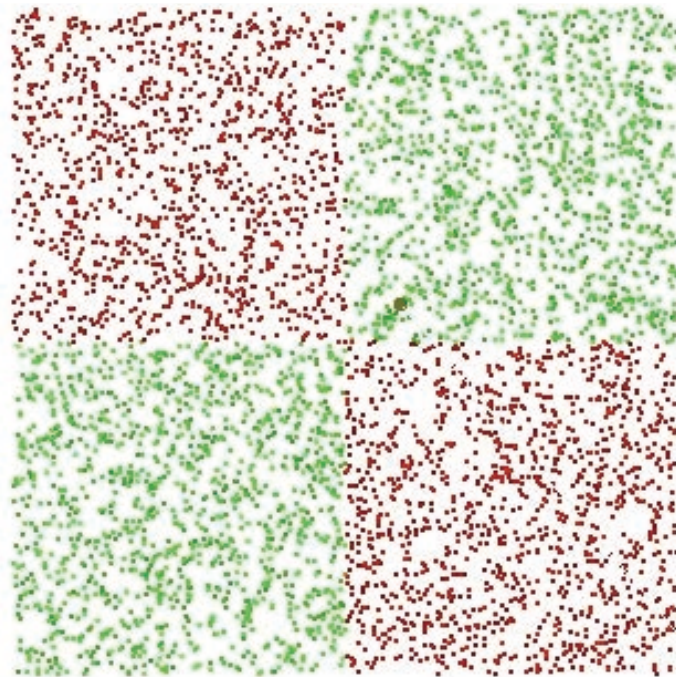
With Related importance

$$w = \underline{(1, 1, 1, 0, 0, 0)}^T$$

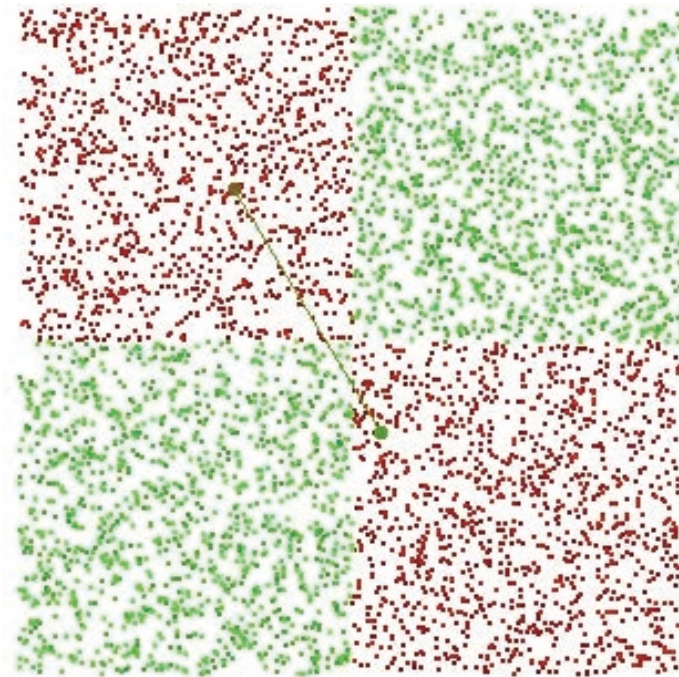


各属性の重みをどう設計するかは非常に難しい問題である

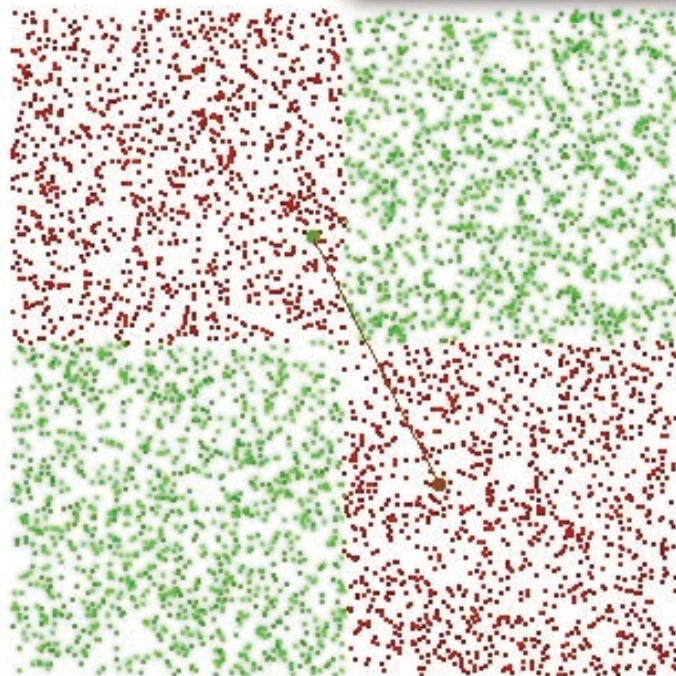
関連重要度を用いた色付き点群の学習



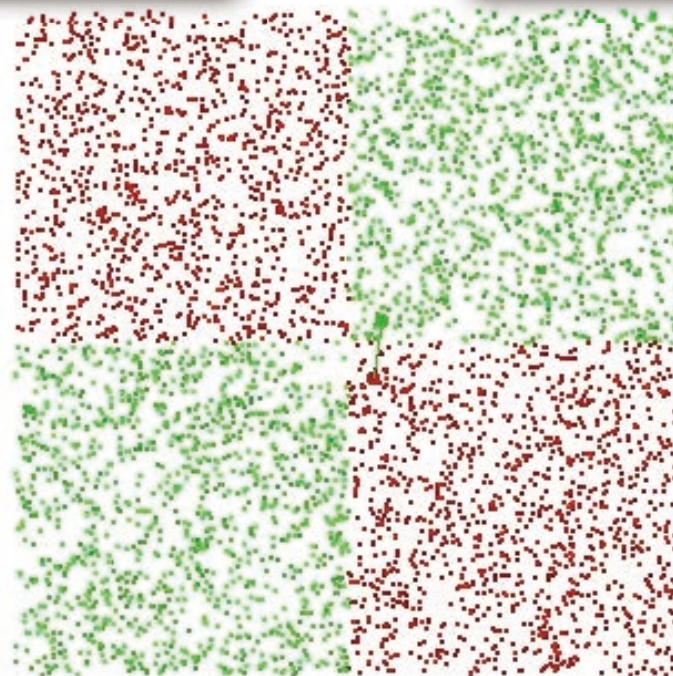
位置:1.0 色:1.0



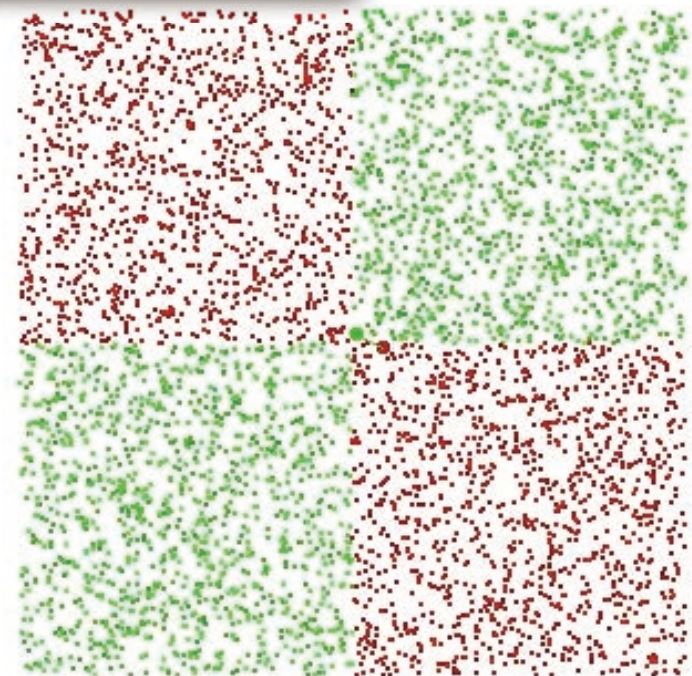
位置:1.0 色:0.0



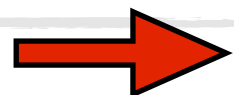
位置:0.8 色:0.2



位置:0.5 色:0.5

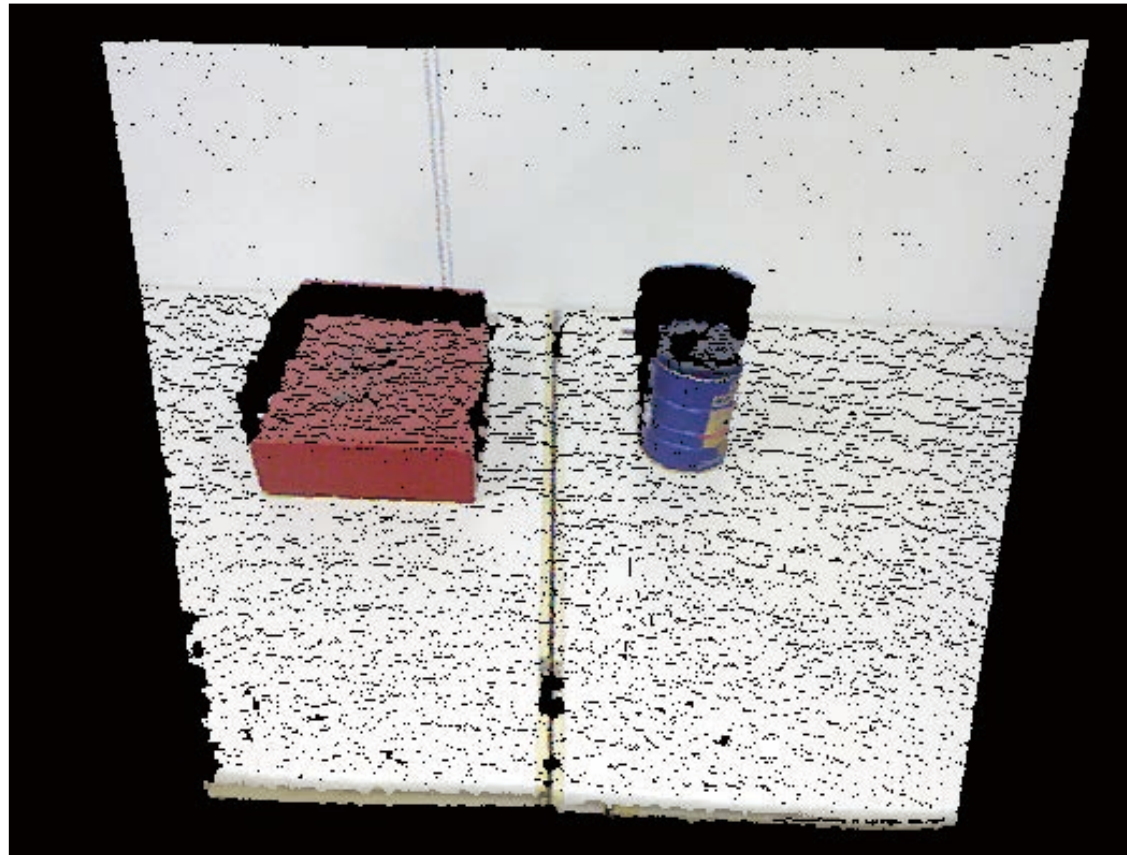


位置:0.0 色:0.0



各属性の重みをどう設計するかは非常に難しい問題である

GNG with Different Topologies (GNG-DT)



3D point cloud

空間

部屋の一部（クラスタ数:1）

色

赤, 青, 白（クラスタ数:3）

形状

4つの面, 円柱（クラスタ数:5）

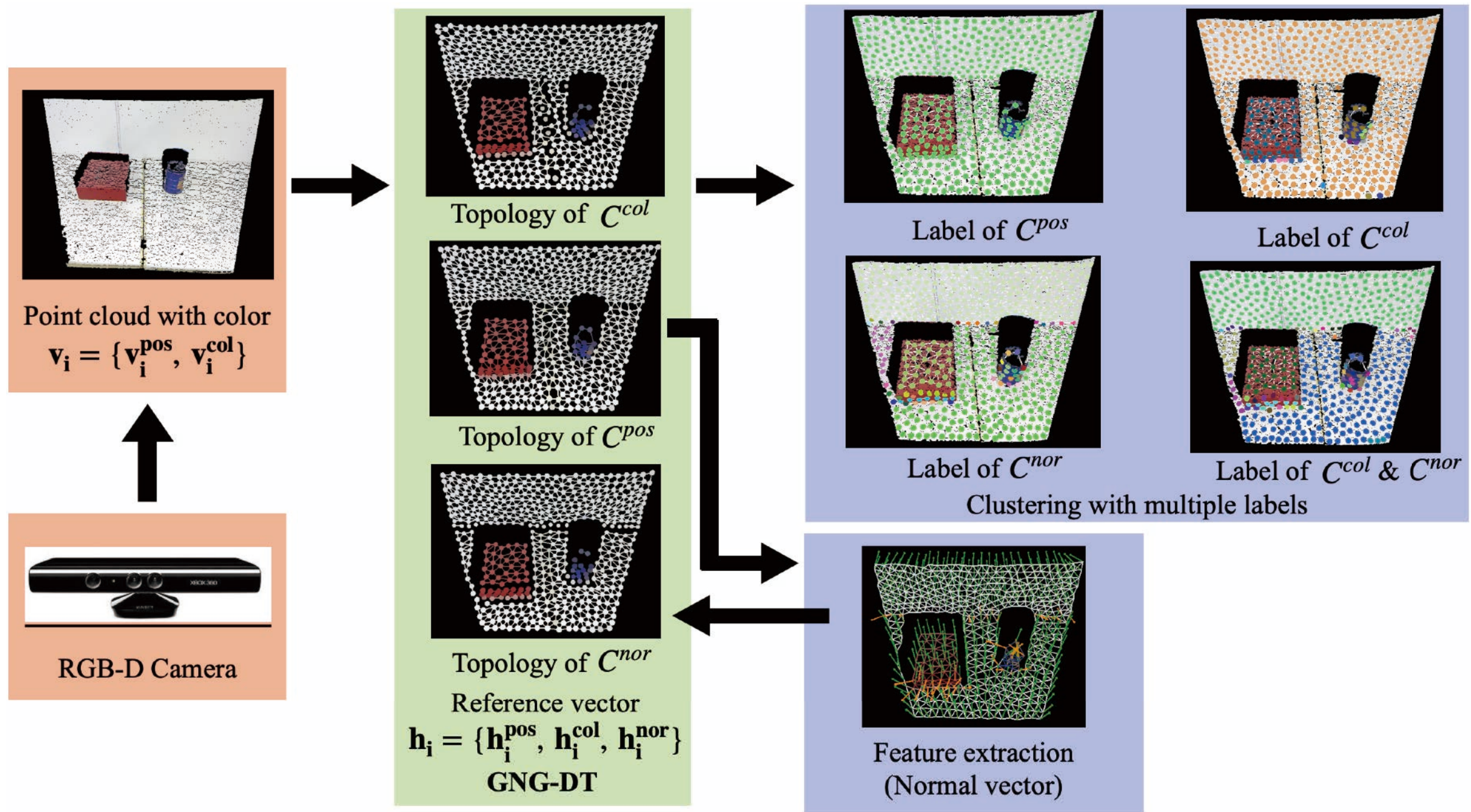
オブジェクト

箱, 缶, 机, 壁（クラスタ数:4）

・
・
・

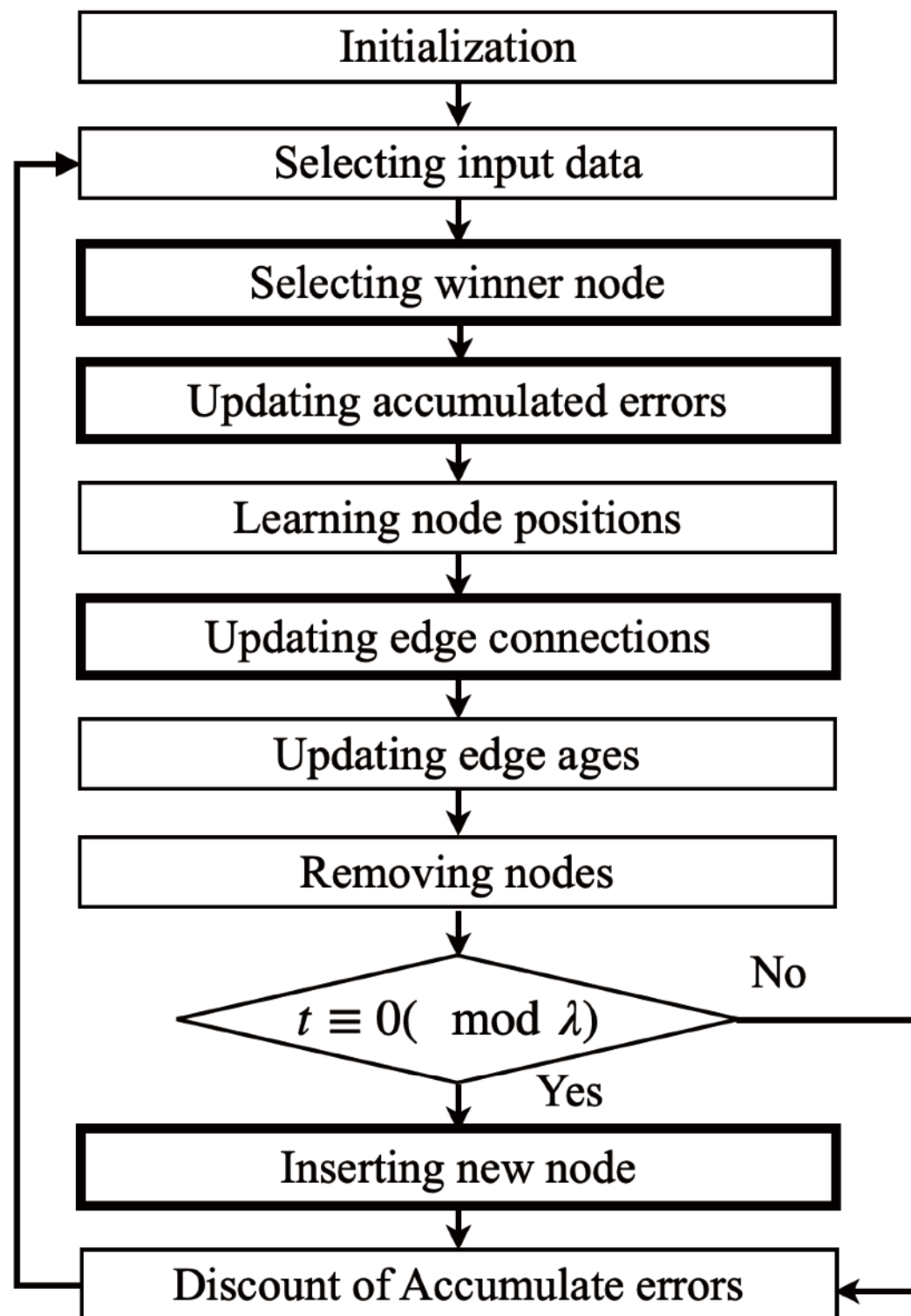
RGB-Dカメラから計測された未知の点群データをクラスタリングする際には、
物体の色や形状などの属性情報ごとに得られるクラスタの結果が異なる

GNG with Different Topologies (GNG-DT)



従来のGNGにおける単一の位相構造の構築ではなく、複数の位相構造を保持による点群処理のための新たな知覚システムの構築

GNG with Different Topologies (GNG-DT)



変数定義

入力ベクトルの属性: $S^{in} = \{pos, col, \dots\}$

参照ベクトルの属性: $S^{ref} = \{pos, col, nor, \dots\}$

入力ベクトル: $\mathbf{v} = (\mathbf{v}^{pos}, \mathbf{v}^{col}, \dots)$

参照ベクトル: $\mathbf{h}_i = (\mathbf{h}_i^{pos}, \mathbf{h}_i^{col}, \mathbf{h}_i^{nor}, \dots)$

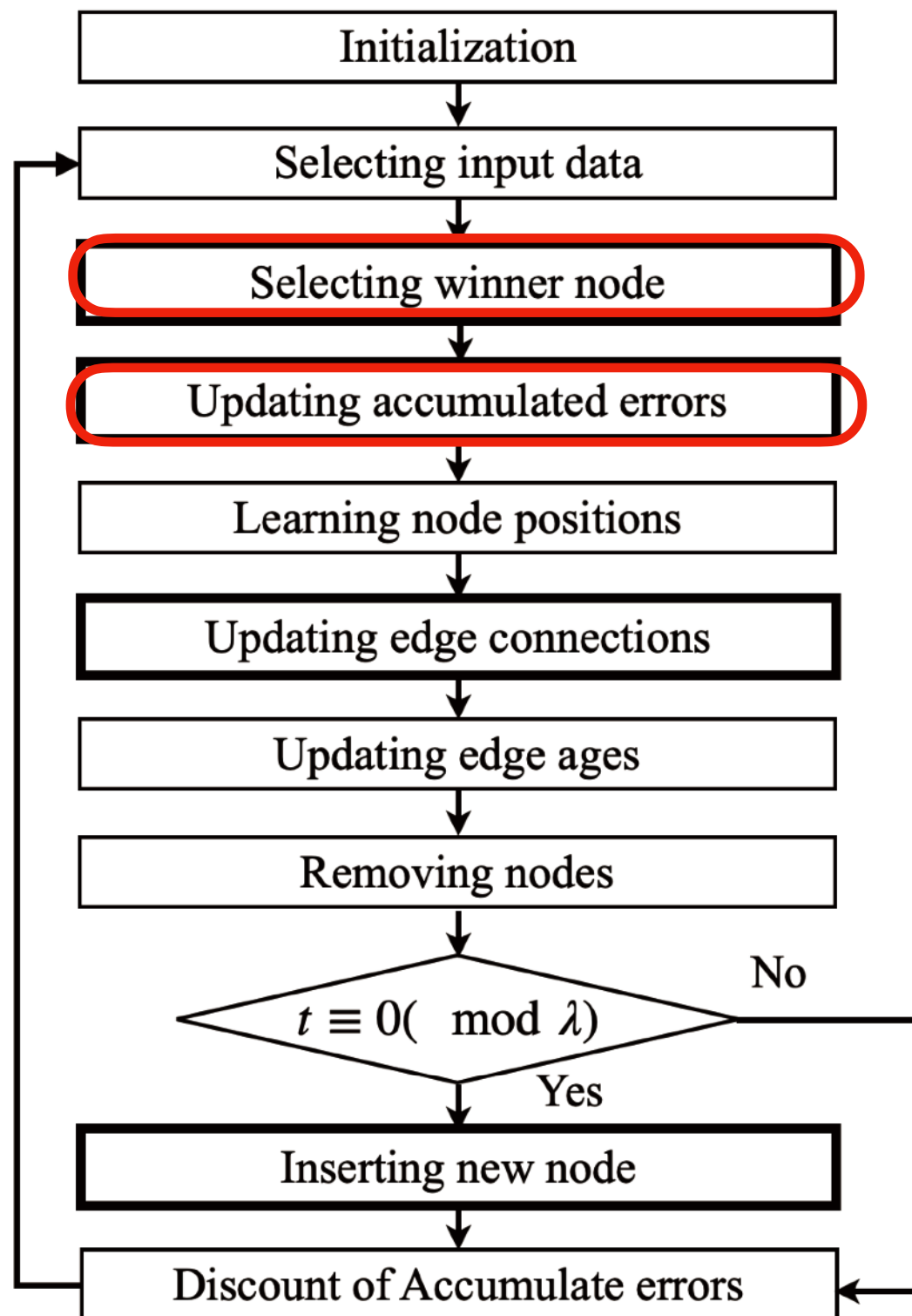
属性 o の位相構造: $C^o = \{c_{1,2}^o, \dots, c_{i,j}^o, \dots\}$

距離定義

属性 o における入力ベクトルと i 番目の参照ベクトルとの距離を以下のように定義する.

$$d_i^o = ||\mathbf{v}^o - \mathbf{h}_i^o||$$

GNG with Different Topologies (GNG-DT)



距離尺度

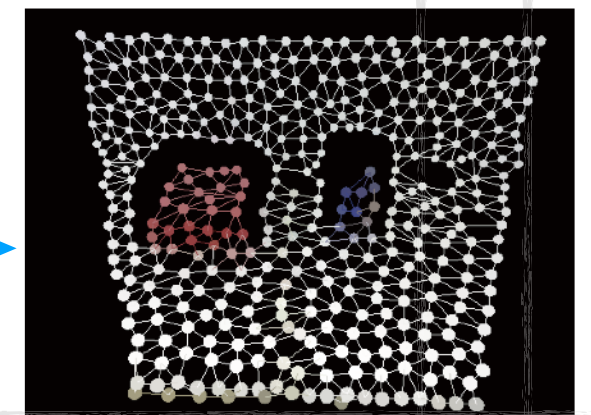
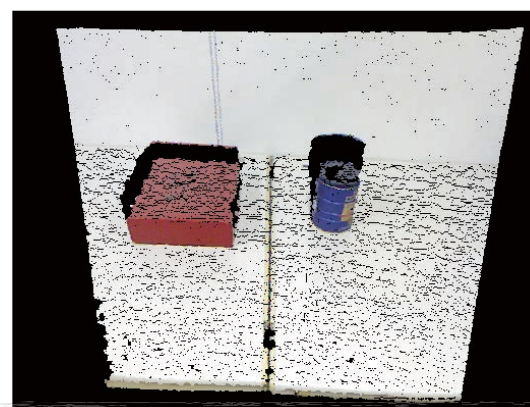
点群の学習では、**幾何的な構造**を保持することが、最も重要な役割となる。
よって、GNG-DTでは勝者ノードと積算誤差の累積における距離尺度として、位置情報(pos)のみを用いる

勝者ノードの選択

$$s_1 = \arg \min_{i \in A} d_i^{pos} \quad s_2 = \arg \min_{i \in A \setminus s_1} d_i^{pos}$$

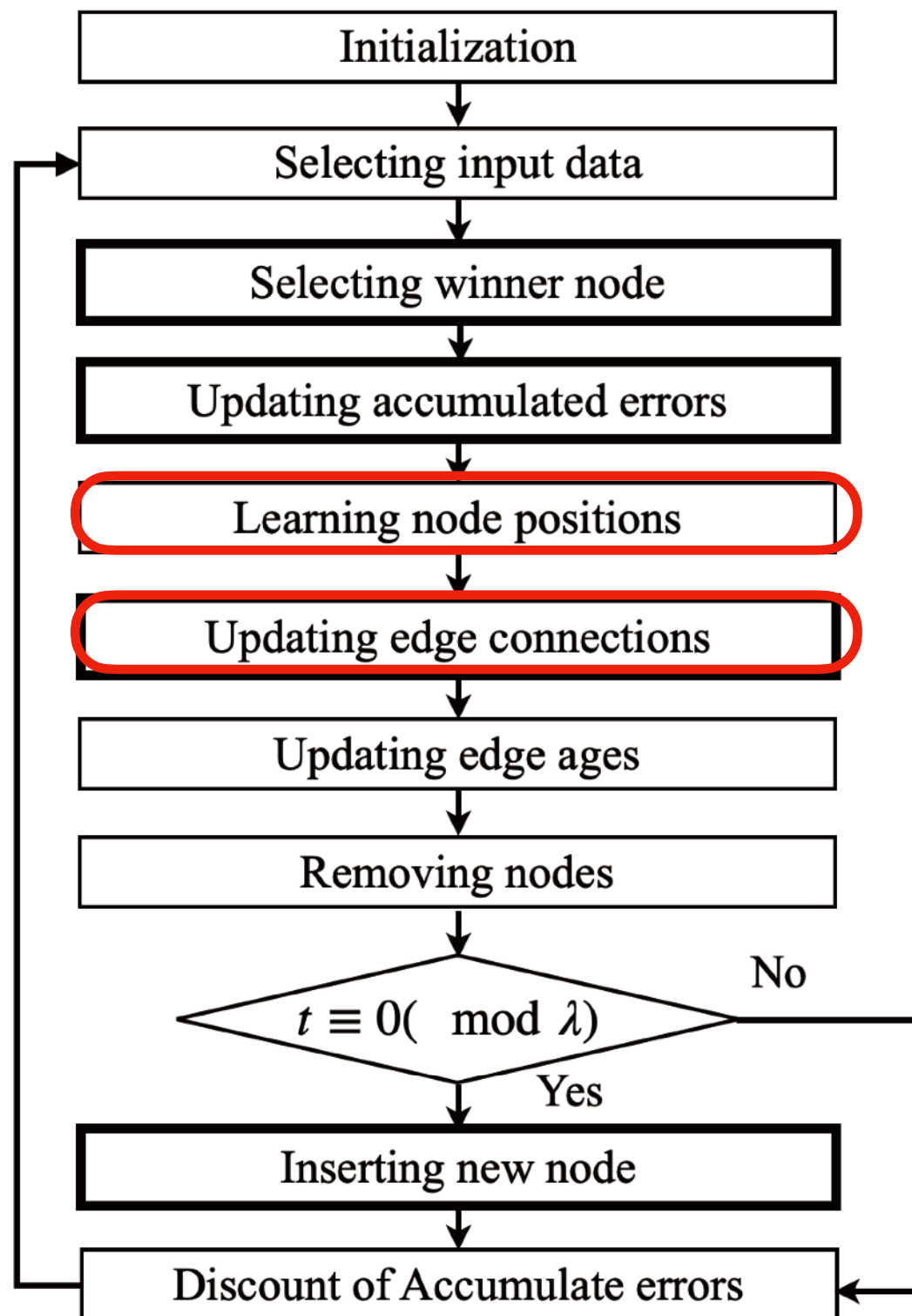
積算誤差の累積

$$E_{s_1} \leftarrow E_{s_1} + (d_{s_1}^{pos})^2$$



位置:1.0 色:0.0

GNG with Different Topologies (GNG-DT)



ノードの更新

第1勝者ノード s_1 は従来どおり更新し, 近傍ノードは各属性の位相構造を利用し更新する

$$\mathbf{h}_{s_1} \leftarrow \mathbf{h}_{s_1} + \eta_1 (\mathbf{v} - \mathbf{h}_{s_1})$$

$$\mathbf{h}_j^o \leftarrow \mathbf{h}_j^o + \eta_2 (\mathbf{v} - \mathbf{h}_j^o) \quad \text{if } c_{s_1, j}^o = 1$$

エッジの更新

位置情報

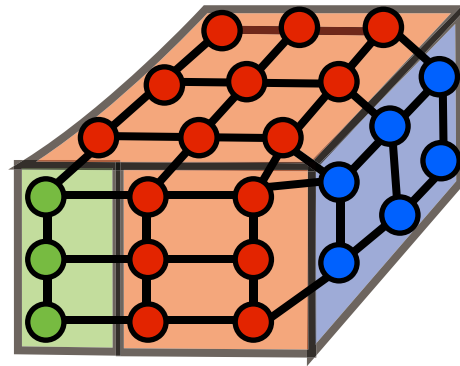
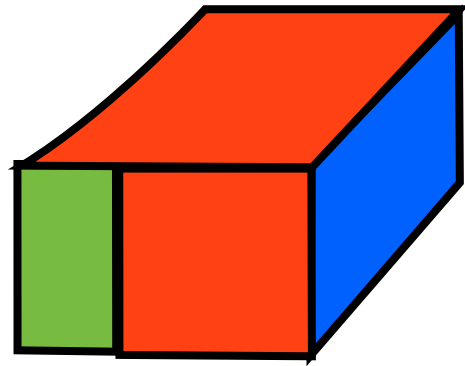
位置情報の位相構造(C^{pos})は, 従来どおり, 勝者ノード s_1, s_2 の間にエッジが存在しない場合, 追加をする.

その他の属性情報

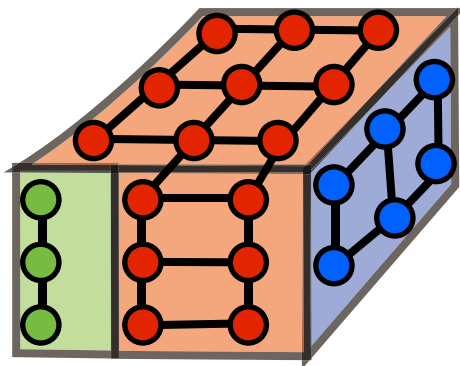
その他の属性情報の位相構造(C^o)は, 以下の式により, 更新する

$$\begin{cases} c_{s_1, s_2}^o = 1 & \text{if } \|\mathbf{h}_{s_1}^o - \mathbf{h}_{s_2}^o\| < \tau^o \\ c_{s_1, s_2}^o = 0 & \text{otherwise} \end{cases}$$

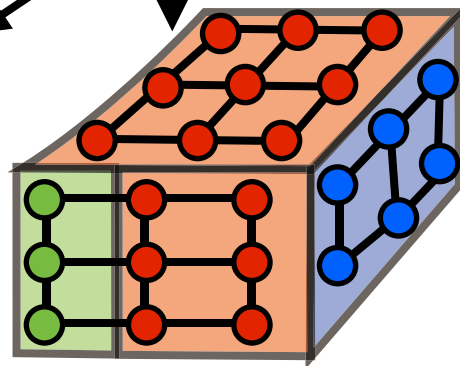
複数の位相構造と組み合わせ



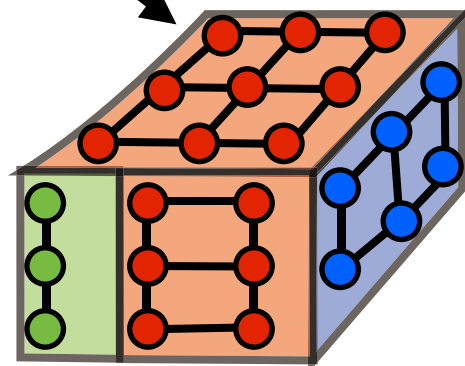
位置情報(C_{pos})



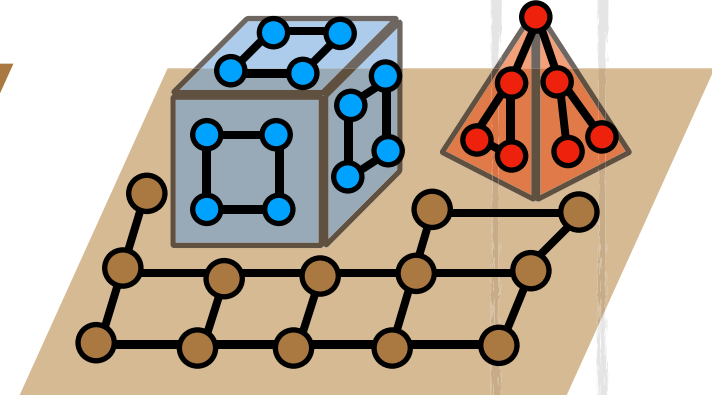
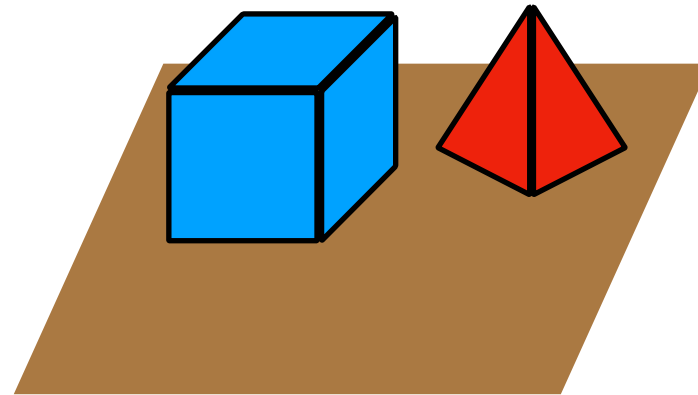
色情報(C_{col})



法線情報(C_{nor})

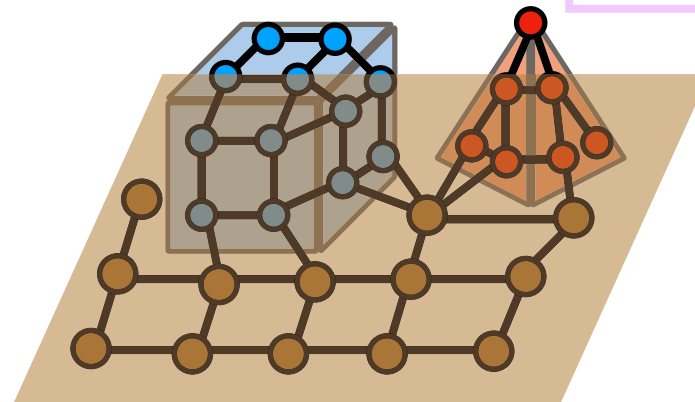


色と法線情報($C_{col} \cdot C_{nor}$)

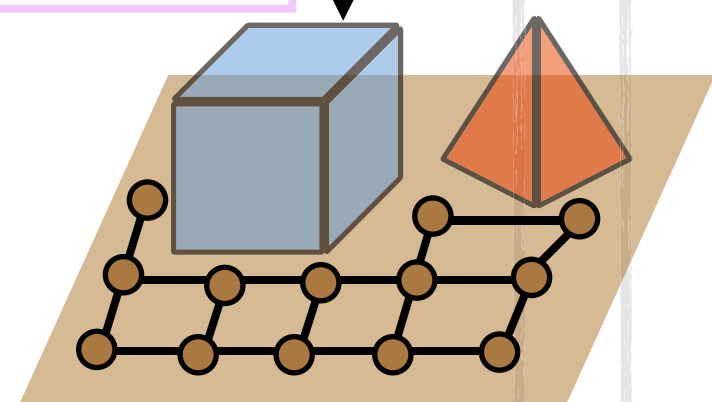


法線情報(C_{nor})

前提知識の利用

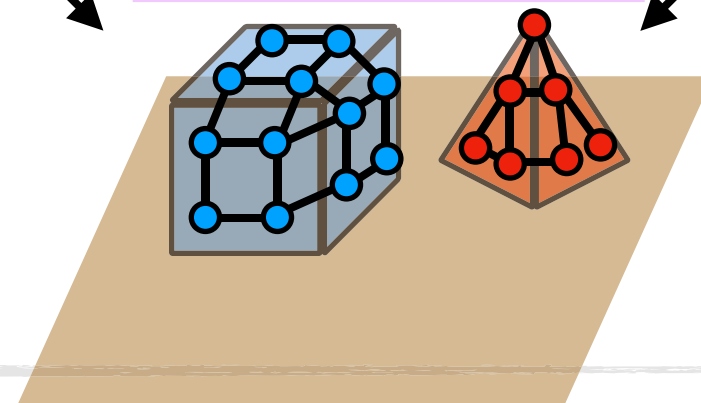


位置情報(C_{pos})



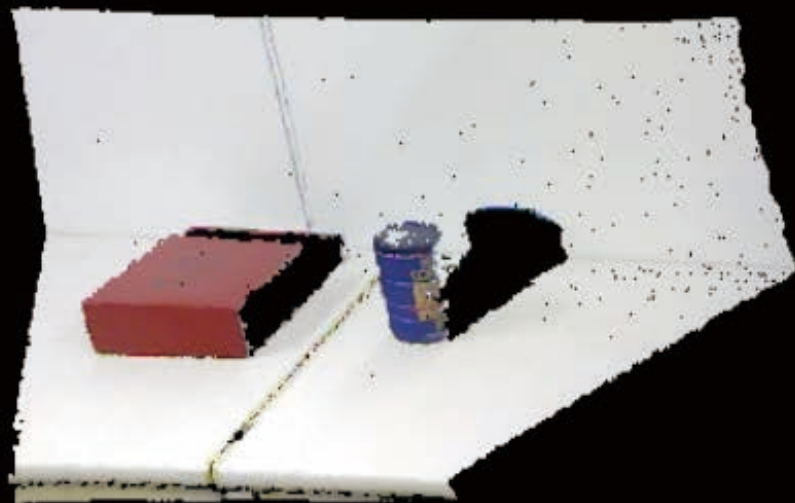
面情報(C_{pos})

ノードの引き算

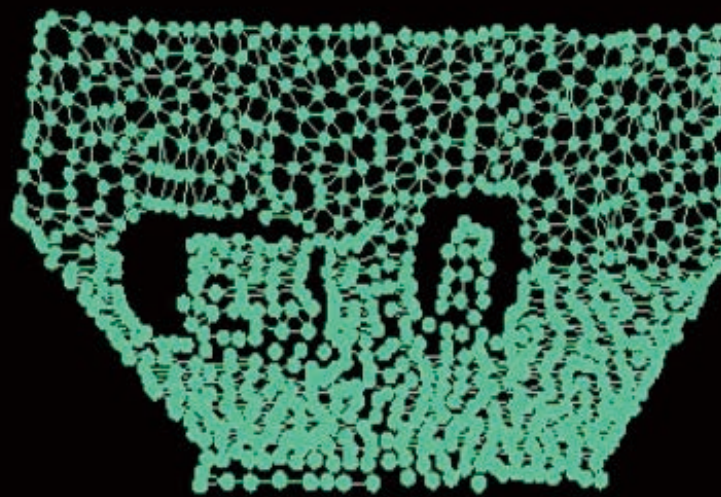


オブジェクト情報

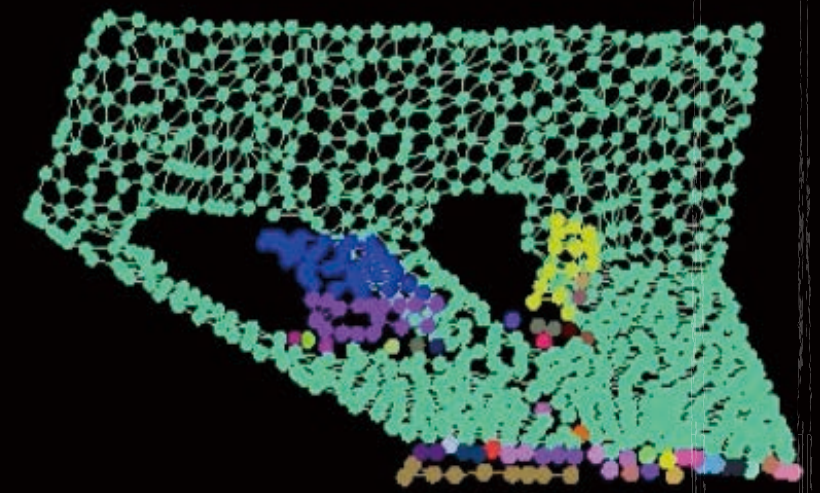
GNG-DTのRGB-Dカメラへの適用



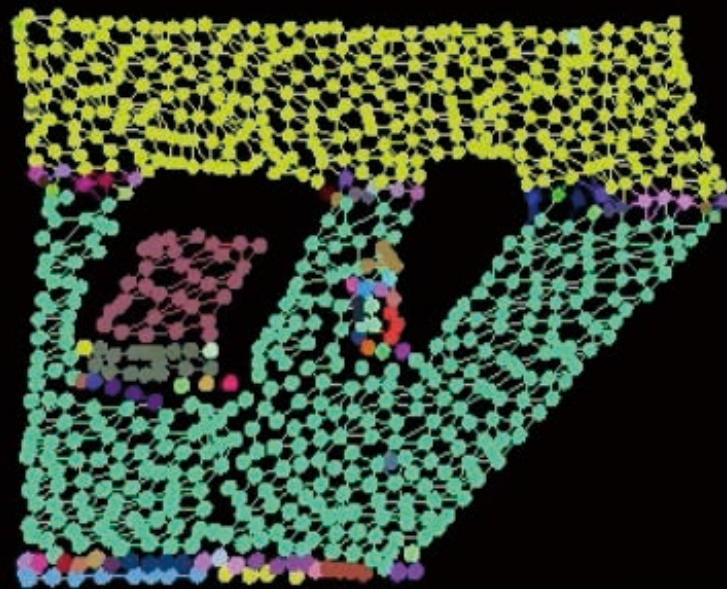
Input Data



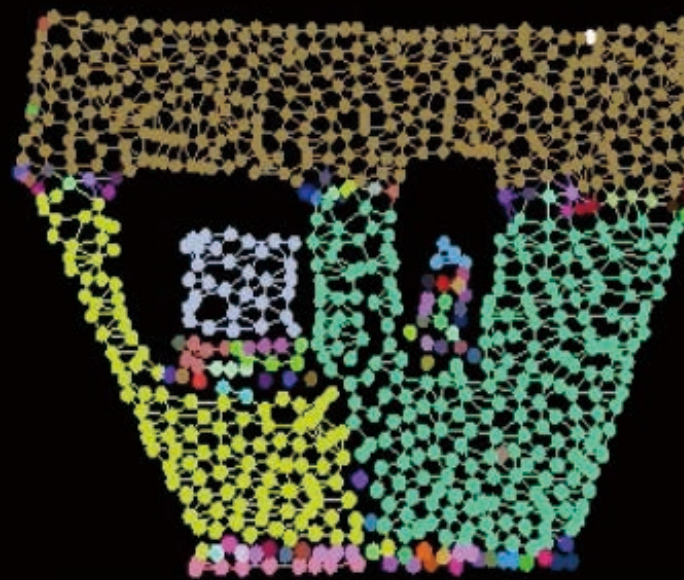
Position Information



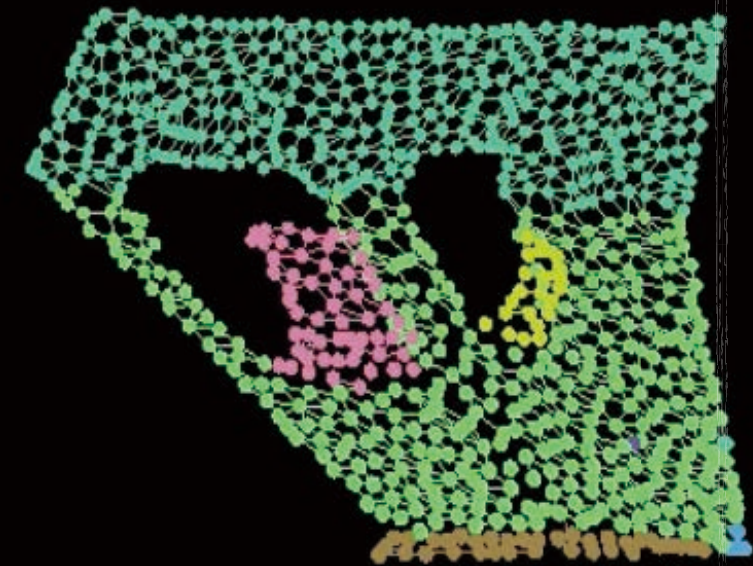
Color Information



Normal Vector Information



Color and Normal Vector Information



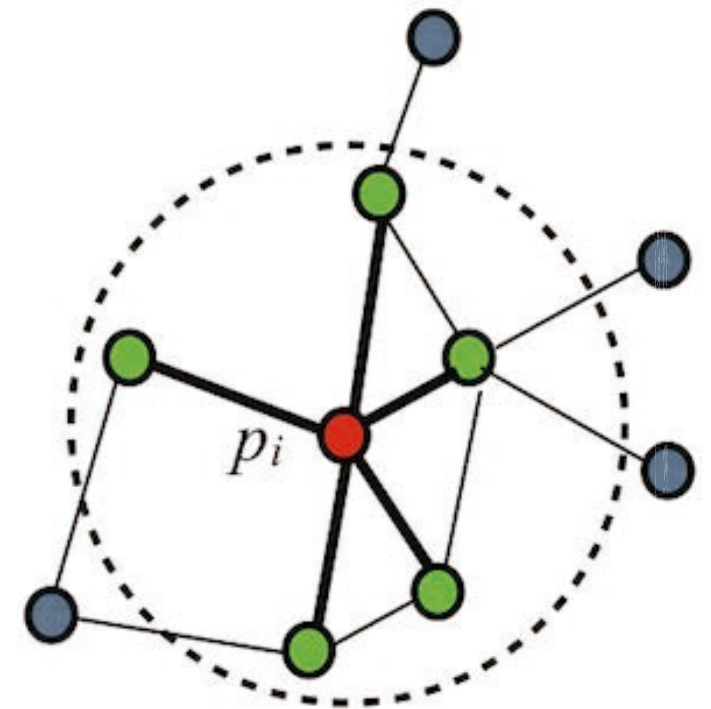
Surface Information

3次元点群における形状特徴量

3次元点群における形状特徴量は、局所的な面要素を構築し主成分分析をすることによって求めることができる。

局所面の共分散行列 C_i の計算

$$C_i = \sum_{j=1}^k (h_j - o_i)^T (h_j - o_i)$$
$$o_i = \frac{h_i + \sum_{j=1}^k w_{i,j} \cdot h_j}{1 + \sum_{j=1}^k w_{i,j}}, w_{i,j} = \exp\left(-\frac{\|h_i - h_j\|^2}{s^2}\right)$$



Local surface

固有ベクトル $\mathbf{v}_j$ と固有値 λ_j の計算

Normal vector $\mathbf{n}_i$

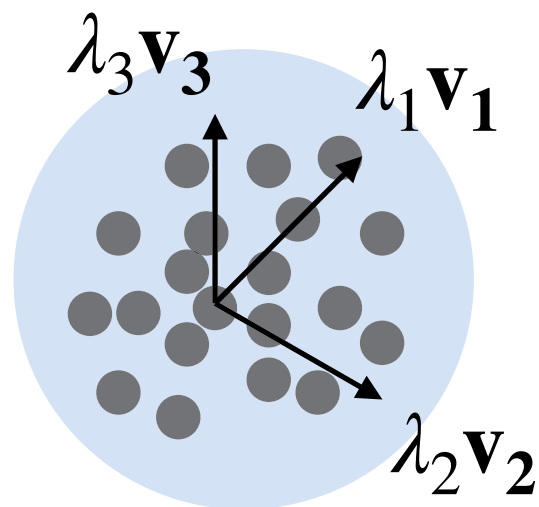
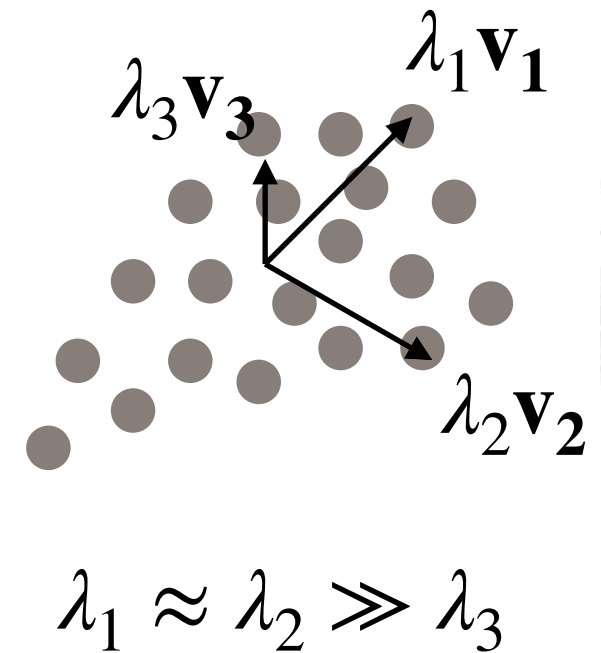
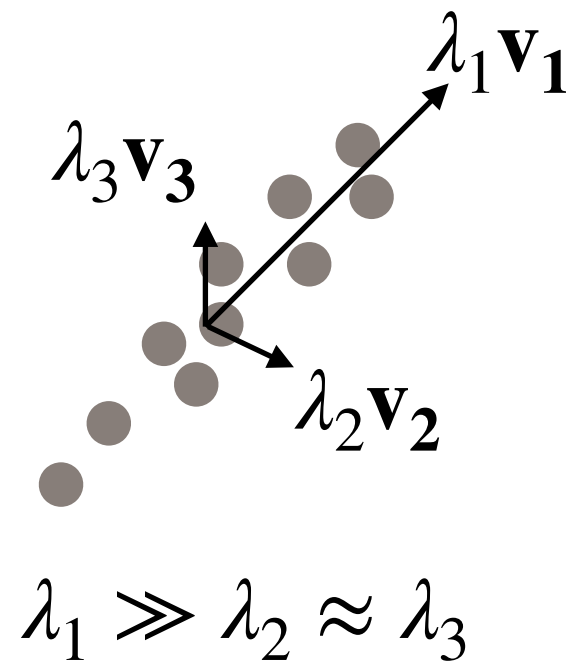
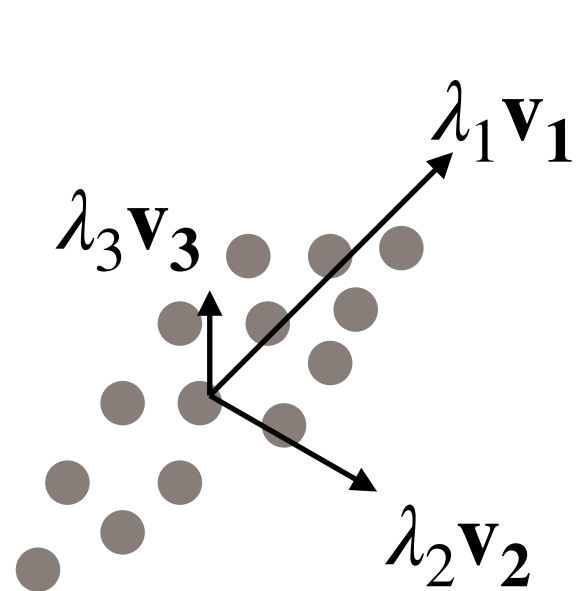
$$\mathbf{v}_3 (\lambda_1 \geq \lambda_2 \geq \lambda_3)$$

Curvature

$$r_i = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$$

3次元点群における形状特徴量

形状特徴量と固有値・ベクトルの関係



$$\lambda_1 \approx \lambda_2 \approx \lambda_3$$

$$s_{1d} = \frac{\lambda_1 - \lambda_2}{\lambda_1} \quad s_{2d} = \frac{\lambda_2 - \lambda_3}{\lambda_1} \quad s_{3d} = \frac{\lambda_3}{\lambda_1}$$